£1.50 ONLY

YOUR AMIGA

December/January 1988/9

£1.50

POW
WHAM
BUY IT! before it's too late
Uurgh...

COMIC SETTER— Let your creativity run riot

Amiga on the box— Create your own TV shows

EXCLUSIVE! MusicX: the ultimate MIDI package

REVIEWED

Star Glider II

Virus

Corruption

Star Ray

SERIES

CLI for beginners

AmigaBASIC

68000 Programming

First Words in C

F-Basic Reviewed

NEW

MICRO BLASTER

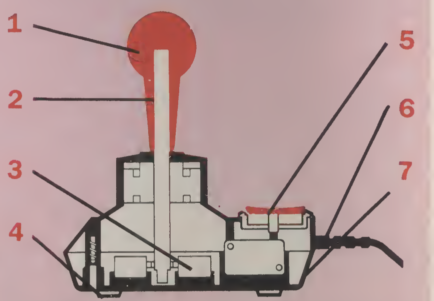
REPLAY® PROUDLY PRESENTS...

The New Microblaster Joystick, professional arcade quality, fully microswitched for the ultimate in game control.

ORDER TODAY AND TAKE CONTROL!

FEATURES INCLUDE

1. Ergonomically designed handle and base for smooth, accurate and comfortable playing action.
2. Steel shaft for extra strength & durability.
3. 8-way arcade quality microswitches giving greater sensitivity and precision controlled movement.



4. Non-slip rubber feet on base for secure table mounted operation.
5. Large, dual-fire microswitched fire buttons for left or right hand operation.
6. Extra long 1.4 metre cable allows more choice of player position.
7. Switchable normal/rapid fire options for extra player control and higher scores.

Suitable for use with a wide range of computers and video game machines.

Compumart Ltd., Jubilee Drive, Loughborough, Leics. LE11 OXS. Tel: (0509) 610444

REPLAY

Compumart are the Officially Appointed Sole UK Dealer for Replay Joysticks.

GUARANTEE
This quality Replay® joystick is fully guaranteed against material defect or faulty workmanship for a period of 12 months from the date of purchase.

ORDER TODAY AND TAKE CONTROL!

PLEASE RUSH ME ___ JOYSTICK(S) @£12.95 each.

Total enclosed £___

Type of computer owned___

Despatched same day. Allow maximum 7 days delivery.

Please deliver to:

Name___ Date___

Address___

___ Postcode___

Price inc. VAT, free postage & packaging U.K. only subject to availability. E. & O.E.

☎ (0509) 610444
24 HR HOTLINE
to order using
ACCESS/VISA
or complete the
coupon and send with
a cheque/ P.O. to:

Access
VISA
REPLAY

COMPUMART LTD
DEPT. (YA)
FREEPOST
LOUGHBOROUGH
LE11 OBR

Editor: Stuart Cooke
Deputy Editor: Fin Fahey
Group Editor: Mark Webb
Ad Manager: Paul Kavanagh
Ad Copy Control: Andrew Selwood
Origination: Project 3
Designer: Kim Goodhew

Your Amiga is a bi-monthly magazine appearing on the third Friday of every second month. The contents of this publication including all articles, designs, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the written consent of the company. ©1988

Distribution SM Distribution, 6 Leigham Court Road, London SW16 2PG. Printed by Redwood Web Offset, Trowbridge, Wiltshire. Origination: Project 3, Whitstable, Kent.

Your Amiga, Argus Specialist Publications Limited, Number 1, Golden Square, London W1R 3AB. Telephone: 01-437 0626 Telex: 8811896. Opinions expressed in reviews and the opinions of the reviewers are not necessarily those of the magazine. While every effort is made to thoroughly check items published for errors we cannot be held responsible for any mistakes that do occur. We regret that we are unable to answer readers letters on an individual basis.
Subscriptions are available from: Infonet Ltd, 5 River Park Estate, Berkhamsted, Herts, HP4 1HL. Rates are as follows: UK £9.00, Europe £11.80, Middle East £11.90, Far East £13.00, Rest of the World £12.10.
Above rates are for surface mail. For airmail rates please contact Infonet at the above address. US subscriptions by Wise Owl Publications, 4314 West 238th Street, Torrance, CA90505 USA.

This month's cover by Alan Batchelor using *Comic Setter*.

ARGUS
PRESS
GROUP

Welcome

■ **Eight months can be a very long time in the world** of computers. It's just that long since *Your Amiga* erupted on to the newsstands, with the intention of fulfilling those readers' needs that other, and perhaps less fresh, Amiga magazines can't meet. We like to think that we've succeeded.

Fact is, the Amiga is an awful lot more than an Atari ST clone, which means a games console with a keyboard. Even the ST is undeserving of this fate — it was once a reasonable machine for the price, but now seems to be consigned to a leisure limbo from which there may be no recovery.

No-one has any gripes about software houses knocking out heaps of games — quite the opposite in fact — but for a machine to enjoy a really healthy life in the market, it must have a user base that is as broad as possible.

Look at the good old C64. A superlative games machine for its time, which not only became the mainstay of many a small business, but spawned a generation of expert machine-code hackers as a result of its extraordinarily fluid design.

Which brings us back to YA. We see the Amiga as the C64 of tomorrow. The problem with this machine, though, is its sheer power. It's a whole order of magnitude more perplexing than its older stablemate, and several times more complex than the ST. This means that the Amiga is a machine that is going to take a lot of time, patience and sheer dogged cussedness to come to terms with as it struggles to mature.

We hope that we're easing the growing pains. Our technical coverage of the Amiga is reaping an increasingly favourable response from readers striving to move on from Basic and 6502/Z80 programming to the bewildering world of the 68000, C and above all the scintillating complexity of the Amiga's internal architecture.

As the year draws to a close, the Amiga is rapidly gaining ground against its major competitors though this is possibly due, in part, to chip shortages that have forced other companies to jack up their prices.

A whole new generation of users are going to be hacking into the machine, you may be one of them. If so, remember, you're not alone while we're here.

Stuart Cooke

Contents

FEATURES

Strictly for Beginners 14

Comic Setter 24
Do-it-yourself Superheroes

Improve your Image 40
The latest in video production software

Music X 50
Exclusive review of the all-singing music package

F-Basic 64

ENTERTAINMENT

Starglider 2 12

Empire 22

Ultima IV 26

Star Ray 31

Corruption 39

Powerplay 57

Lancelot 59

Bomb Busters 61

The Empire Strikes Back 62

Virus 66

Zoom 67

Space Racer 68

PROGRAMMING

First Words in C 16

CLI 20
More on the command line interface

Amiga BASIC for beginners 28

Amiga Lib 34
How to use the Amiga's routine libraries

Corrupt Practices 44
Repairing your disks — and more . . .

Assembly Language Programming 54
Getting started in 68000

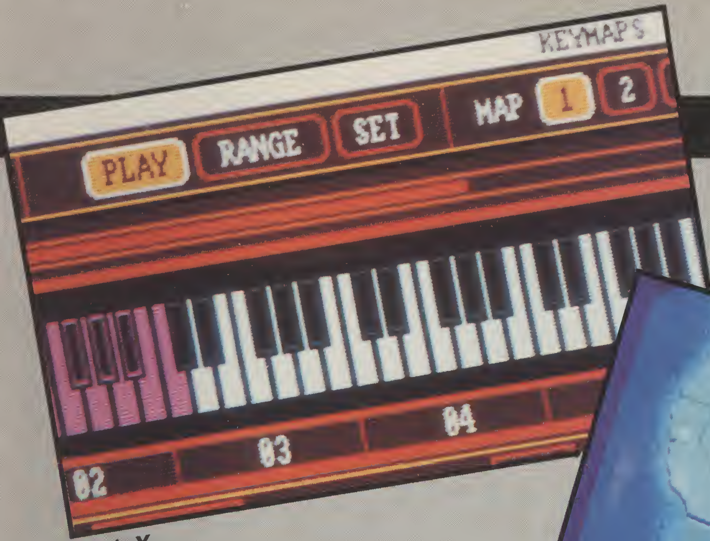
REGULARS

Welcome 3

Amiga Update 9
The latest goings on in the world of Amiga

Guru's Guide 47

CONTENTS



Music X

Deluxe Productions

Virus

Starglider 2

Comic Setter

Empire Strikes Back

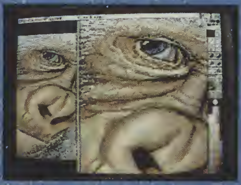
All software is designed especially for the Amiga computer.

DELUXE CREATIVITY HAS ITS LIMITS.

YOUR IMAGINATION.

The Deluxe Series of high quality creativity tools from Electronic Arts.

A set of totally integrated packages that is so powerful and so flexible, it is limited only by your own imagination.



DeluxePaint II with DeluxePrint I The world standard DeluxePaint is now unbeatable with the inclusion of DeluxePrint. With so many built-in artistic tools, they'll bring out the artist in you!

DeluxeMusic The ultimate music tool for composing, performing and printing. DeluxeMusic features an enormous variety of playback options including Midi I/O.

DeluxePhotoLab Combines three powerful graphics tools – a professional paint program that works in all Amiga graphics modes including H.A.M., a colour image processor and a poster maker for *giant* posters!

DeluxeVideo Rock videos, business graphics, cartoons . . . this comprehensive video animation package allows you to make your own stunning videos, without complicated commands and formats.

DeluxePrint II What you see is what you get. DeluxePrint II has seven ready-made page formats for signs, banners, cards, letterheads, calendars and more!

Available now at your nearest Amiga software outlet. More information is available from:
Electronic Arts, 11/49 Station Road, Langley, Berkshire SL3 8YN. Telephone (0753) 49442.

ELECTRONIC ARTS
Deluxe Creativity Series

Charon 5 is an unparalleled design concept that combines strategy with Arcade action to produce a game of epic proportions. The action takes place on over 5,400 full color screens spread throughout 11 action packed levels. The game is further enhanced by the use of exceptional true stereo sound and music.

You must exhibit skill, have concentration, possess good reflexes and not be faint of heart in order to be successful on your mission against Charon 5.

Mindware International

Precision Distribution
6 Park Terrace
Worcester Park
Surrey KT4 7JZ
Tel: 01–330 7166
Fax: 01–330 2089

For more information or the dealer nearest you call: **01–330 7166**

Order line: **01–330 7166**

Precision Distribution

£19.95 (inc VAT)

Amiga is a trademark of Commodore Business Machines, Inc. Charon 5 is a trademark of Mindware International.

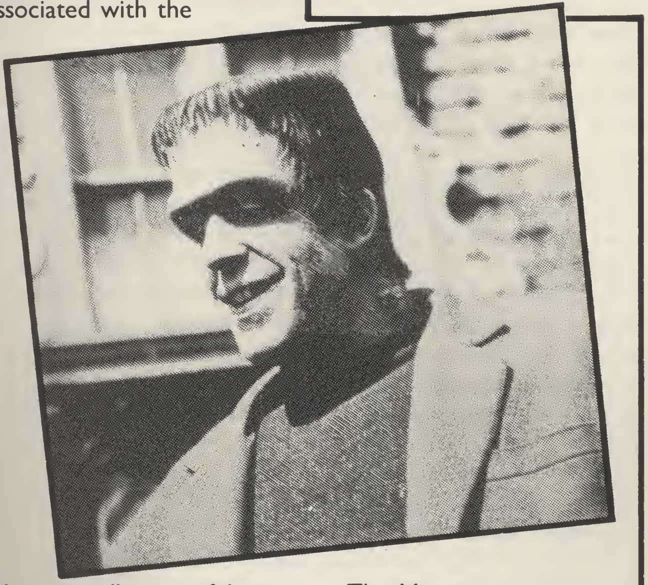
AMIGA UPDATE

Munster Fun

The reshowing of the mid-sixties comedy series, *The Munsters*, has been hailed by one woman's magazine as being 'in' and the current cult status has given birth to a computer game from Again Again, featuring Herman, Lily and the rest of the weird family.

The programme concerns a typical family with two parents in their mid-thirties, typical except for one thing, Herman is a Frankenstein monster and Lily is a vampire. Their ten-year old son Eddie is a werewolf who loves wargames and the extended family also includes a 400-year old vampire grandfather and Marilyn. Marilyn is Herman's niece and the oddball of the family because she is strikingly normal.

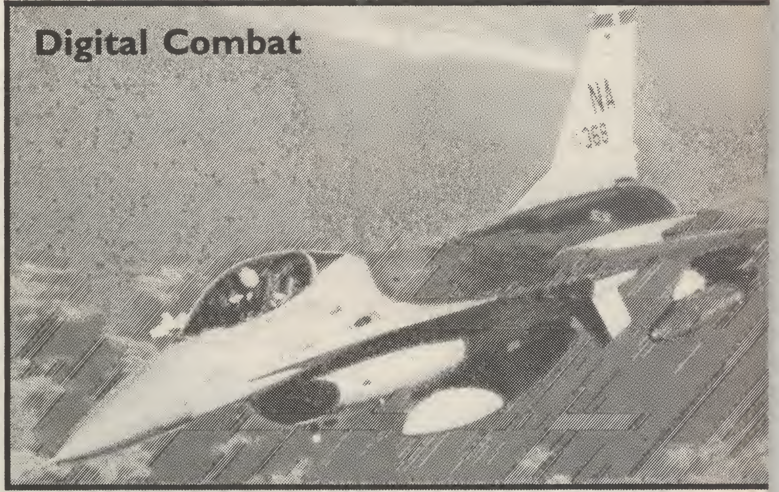
Again Again is a new label associated with the phenomenally sucessful Alternative budget software stable and the Rotherham-based programming team, Teque, is responsible for the program concept and design. Teque are becoming hot property through games like *Blasteroids* and *Pac-Mania* which they are producing for established software houses like Image Works and Grand Slam Entertainments.



The Munsters computer game is an arcade adventure which concerns the kidnapping of Marilyn and the Munsters' attempts to find her. Even the pet dragon who lives under the stairs is brought out for an airing in the game.

As yet, no price has been fixed for the game but it will be released in early 1989.

Digital Combat



■ **Digital Integration is** hoping that *F-16 Combat Pilot* will take off in a big way this Christmas. Its four-man team has been working for over a year on the detailed project, interviewing F-16 pilots and consulting aviation experts to ensure that the program is as accurate a portrayal of the real multirole fighter as possible.

The package is actually a compendium of several games ranging from training flights to a full strategic battle scenario. Additionally, there is a two-player dogfight which is played by connecting two Amigas together.

Simulating the latest F-16C, the software takes a novice pilot through all five of the main roles that a modern pilot must master to become a Squadron Commander. The various missions certainly appear to be comprehensive with the five main areas being identified as air-to-air combat, battlefield support, offensive counter-air attacks on landing strips and radar or missile sites, reconnaissance flights and inderdictor assaults on factories and power stations.

Pilots who prove their command of the full range of strikes will be promoted to plan a campaign against the enemy using all five squadrons against the enemy in a combined offensive.

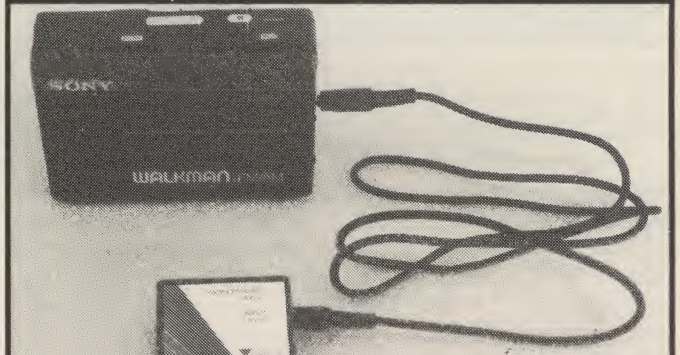
The technical specifications for the game are impressive with each battle area covering 20 000 square miles, a potential 1 500 targets within each area and a different mission each time a pilot takes to the air.

F-16 Combat Pilot should be released early in 1989 and the cost will be £24.95.

Sound Device

Trilogic has produced an audio digitiser which it is claimed offers superb performance but the price is only £24.99. The Amiga Audio Digitizer plugs into the parallel port of the computer and connects via a minijack lead to a personal cassette, hi-fi system or similar sound source. The input has an adjustable level backed up by an overload indicator which is designed to ensure optimum results every time.

The digitiser is not supplied with software but has been designed to comply with the requirements of programs such as *Audiomaster*, *Prosound* and *Perfect Sound*.



California Dreaming

One would imagine that a computer show dedicated to Amiga computers held at a fashionable location on the west coast of America would attract a large amount of attention. Well, you would be both correct and incorrect in your assumption. The software and hardware manufacturers were certainly in abundance, most of the big names were there. Many companies who didn't have stands had people walking around the show arranging deals and press coverage.

MIDI Magic

Unfortunately the American public did not seem to greet the show with the same amount of enthusiasm as the manufacturers. Even on Saturday, the say when most shows are at their busiest, there was a relaxed atmosphere and none of the usual hustle and bustle. Perhaps the fact that the temperature in LA during the show was over the 100 degrees mark had something to do with this apparent lack of attendees? After all who wants to play with computers when the sun is shining and the beaches call?

As you would expect AmiExpo saw the usual new product announcements. A few of them are detailed below: **Brown Waugh,** *16795 Lark Ave, Los Gatos, CA 95030*, had a treat in store for midi musicians with the launch of **Midi Magic**. Described as a "Magical Music Sequencer" the producer offers multichannel simultaneous recording, a resolution of 480 PPQN, real time and step recording and much more. The program has an extensive on line help function and makes full use of the Amiga's window and menu facilities. Up to 26 sequences can be recorded using a maximum of 16 tracks.

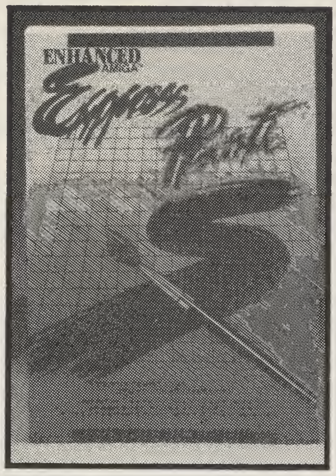
Brown Waugh was also previewing *Express Paint version 3.0*. This version of the program includes 3-D perspective manipulation and unlimited undos. A virtual page facility allows the user to produce a picture that is much bigger than the screen. To get the most from *Express Paint 3.0* it does look as though you are going to need a machine with more than 512K of memory. The press demonstration was running on a standard A500 and the memory limit was reached every few seconds.

Aegis Developments, *2115 Pico Blvd, Santa Monica, CA 90405*, were extolling the virtues of their newly released *Aegis Draw 2000*. This program replaces *Aegis Draw Plus*, one of the most popular CAD programs for the Amiga. Prices at $279.95 the program requires as least 1 M of memory to run.

One feature of the program that will please advanced users is its ability to use the 68020/68881 coprocessor board, Aegis claim an increase in speed of up to 70 per cent if the coprocessor is used.

Audiomaster II was also being previewed. This version of the extremely popular sound sampler features digital sampling rates of up to 56K per second and full stereo support. The software works independently of hardware, and may be used with any sampling device that uses either the Amiga parallel or joystick ports. Registered users of Audiomaster will be able to upgrade for just £30.00. New user prices to be announced.

For some reason sound sampling on the Amiga seems to be one of the most prominent features of the show. **Starvision International,** *305 Madison Ave, Suite 411, New York, NY 10165*, were also demonstrating their new sound sampler. The sampler offers a sampling rate of up to 28KHz, separate left and right channel controllers, Mono/Stereo switch, VU and PEK indicators and comes with all necessary cables and connectors. Price $125.00.



Express Paint

Starvision can probably claim the right to the most tasteless item being shown at the show, the *Terminator* joystick. This joystick, which has been seen in the UK but never took off, is shaped like a hand grenade, a metal rod coming from the top is used as the stick.

All those people waiting to see *Publishing Partner Professional* from **Soft-Logik** will now never see it. For some reason the company has decided to rename the product *Page Stream*. This seems an extremely strange move when we have been hearing about *PPP* for some time and the name is now recognised by anyone interested in DTP on the Amiga. **Soft-Logik** can be contacted at *11131 South Towne Sq, Suite F, St Louis, MO 63123*.

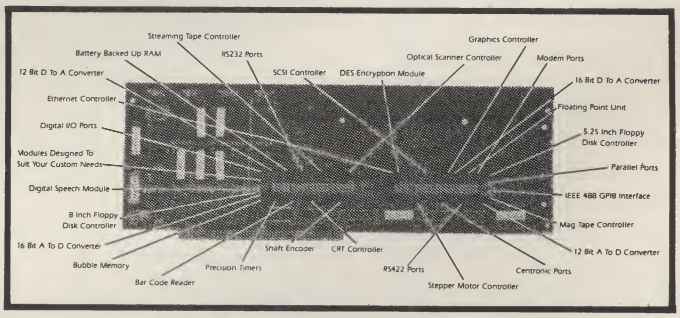
MIcro-Systems Software, *12798 Forest Hill Blvd, Suite 202, West Palm Beach, FL 334514*, originators of the well known productivity software collection the *Works!* announced the launch of the *Works! Platinum Edition*. This package will contain a Wordprocessing Module, Telecommunications, Database, Spreadsheet and the facility to print out your data sideways on your printer. Initial previews of this product look extremely promising. Price $295.

A copy of the first Amiga record, well CD actually, was dropped off at our stand. Called *Gameplay*, the CD features fifteen musical excerpts from popular games. All music is by Jim Cuomo and features excerpts from Defender of the Crown, SDI, Sinbad and many others. For more information contact **Pigeon Music,** *11684 Ventura Boulevard Suite 520, Studio City, CA 91604*.

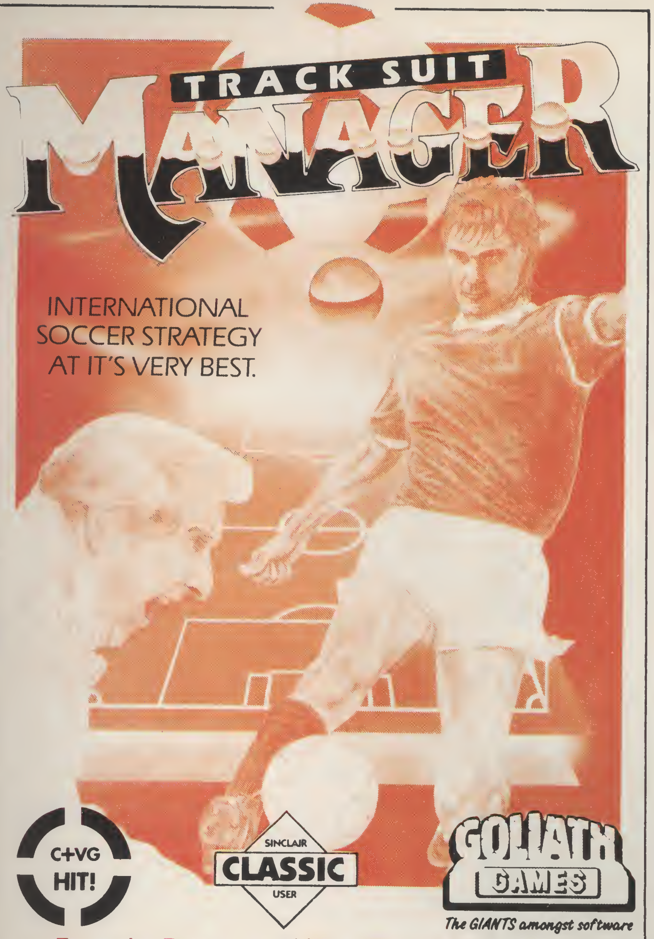
ASDG, *925 Stewart Street, Madison WI 53713*, took the show by storm with their colour picture scanner. SpectraScan is based upon the Sharp Electronics JX-450 colour scanner and ASG's own Twin-X general purpose input/output board which gives the Amiga a high speed IEEE-488 interface bus, among other things.

Unfortunately, as ever, I've run out of space to tell you about everything that was seen at the show. What is important to note is that new software and hardware for the Amiga family is still being launched thick and fast with the quality improving all of the time. We now have Amiga computers controlling what we hear and what we see — what is going to be next?

Stuart Cooke
Ami Expo
LA California



The ASDG I/O Board



COMING SOON – CBM 64. (DISC) £14.95, ATARI ST/ AMIGA £19.95, AMSTRAD £9.95, IBM – P.O.A

From the Designer of 'The Double' comes . . .

TRACK SUIT MANAGER

The first ever football game with real match tactics.

REVIEWS SAY

Commodore User says . . . The match is the best I've ever seen . . . T.S.M. is the most playable game of its genre . . . One of the most frustrating, fun, compelling, exciting and addictive games I've ever played. **8/10**

C & V Games says . . . The best ever football managerial game ever, in the history of the world, no messing, straight up . . . A very impressive start for Goliath. **9/10**

Zzap says . . .
Presentation **96%**
Hookability **93%**
Easily the best of its type
Lastability **88%**
Overall **89%**

Sinclair User says . . . The best ever managerial game, the one your speccy has been waiting for. **93%**

Your Sinclair says . . . I'm amazed at this one . . . Some day all footie games will be made this way . . . If you don't like T.S.M. your dead from the neck up. **8/10**

Your Commodore says . . . T.S.M. is one of the best games I've ever played and certainly highlights the problems of a national team manager

WE SAY

Sophisticated Match Play with ... Adjustable Team Formations ★ Free Kicks ★ Throw In's ★ Offside's ★ Individual Player Abilities ★ Goal Kicks ★ Corners ★ Penalties ★ Tackles ★ Fouls ★ Select Player for 'Set Pieces' ★ Injuries ★ Booking's ★ Sending Off's ★ Suspensions ★ Substitutes ★ Heading ★ Genuine Penalty Shoot Out in Quarter, Semi & Final Matches ★ Extra Time ★

Individual Player & Team (Defence, Midfield, Attack) tactics ... Defensive ★ Attacking ★ Normal ★ Quick Counter Attacks ★ Zonal Marking ★

GOLIATH GAMES is a 'NEW' Software House specialising in ... ADVANCED ... INVOLVED ... ORIGINAL SOFTWARE.

ORDER NOW ... £9.95 Including V.A.T. Plus 50p post & packing

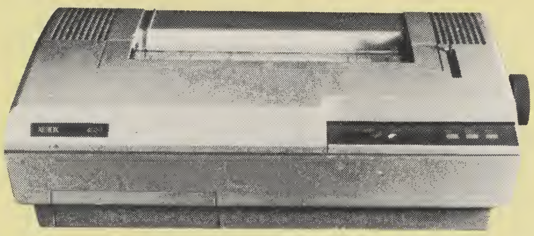
GOLIATH GAMES
The GIANTS amongst software

PLEASE SEND CHEQUES/P.O. TO:-

46 LOCKING ROAD
WESTON-SUPER-MARE
AVON BS23 3DN
0934 622538

COMMODORE 64, SPECTRUM

with the

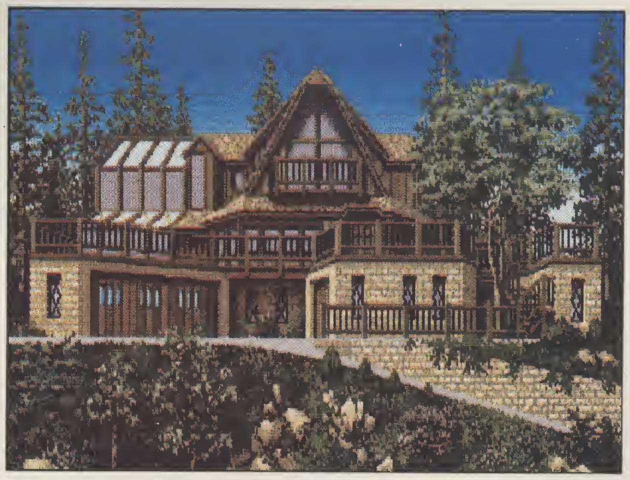


Xerox 4020

Colour Ink Jet Printer

what
you
see

is what you get



With NLQ too!

The Xerox 4020 colour ink-jet printer makes the most of your Amiga colour pictures, printing clear, bright images silently in 7 distinct colours and over 4,000 shades at 240 dots per inch resolution.

The Xerox 4020 handles all your correspondence and desk-top publishing needs too! Choose from 5 resident fonts in various sizes and achieve near letter quality results at a healthy 40 cps.

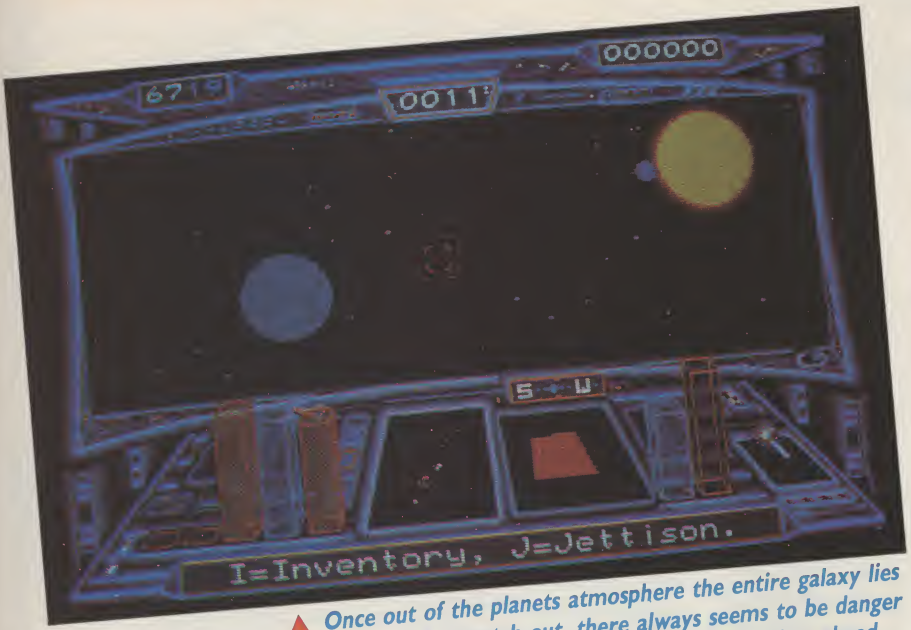
Xerox 4020: the ideal Amiga printer!

Precision
Software

Available from your local dealer or directly from
Precision Software Ltd. 6 Park Terrace,
Worcester Park, Surrey KT4 7JZ
Telex 8955021 Precis G
01-330 7166

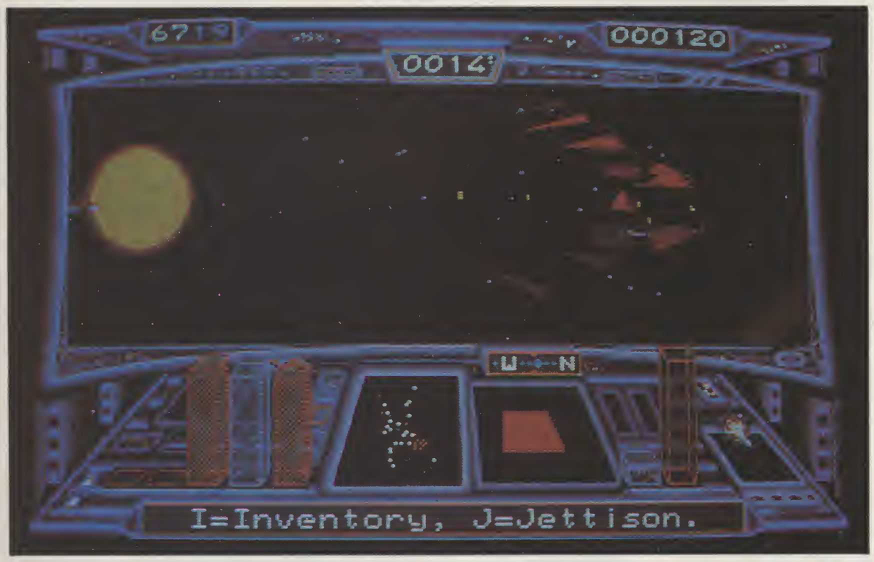
Starglider 2

First person space shoot-em-ups have come a long way since the days of Star Raiders as Kevin Crosby finds out



I=Inventory, J=Jettison.

▲ *Once out of the planets atmosphere the entire galaxy lies before you. But watch out, there always seems to be danger lurking just ahead…*



I=Inventory, J=Jettison.

▲ *It's battle stations as pirates throw some serious photon death in your general direction, but fear not, in this instant good triumphs over evil as the remains (and cargo) of some late inter-galactic Long John Silver bombard your ship. Skilful use of the tractor beam at this point should secure the ex-smugglers' booty*

We've all seen a lot of cockpit view space games over the years. Most notably the likes of *Star Raiders*, *Elite* and *Mercenary*. Surely this particular genre has gone as far as it can? Well, it would appear not.

Starglider 2 is set to push back a few boundaries where perceived realism is concerned (obviously we can't talk about actual realism when dealing with fantasy). On first sight it is obvious that you're dealing with some slick programming, with smooth scrolling, filled-in graphics (not wire frame) and tremendous sound effects.

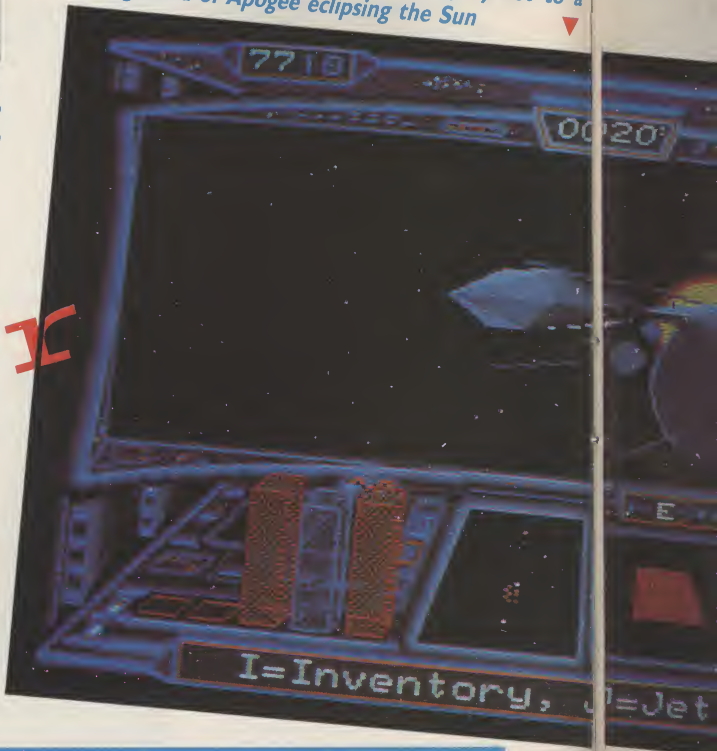
The game itself is a set mission which can be completed in several ways. You have to stop the Ergons from building a great big space station that will blow away all the nice friendly planets in the galaxy. Precisely how this is done I'm not letting on, suffice to say that it involves visiting most of the planets in the galaxy and blowing up an awful lot of stuff!

The guys at Argonaut software have obviously taken on board what makes a good space game like this and rolled most of the good elements of each into *Starglider 2*. The functionality of *Star Raiders* is there along with the exploration/adventure aspects of *Mercenary* plus, of course the planet-hopping, interstellar trading qualities of *Elite* also to be released soon on the same label for the Amiga). With that sort of combination how can you possibly lose?

Icarus in Action

You've been given a prototype police cruiser for this mission codenamed Icarus. The name becomes clear if you try flying too close to the sun. At the start of the game you are flying close to the surface of

Having picked off a few bad guys it's time to hot-foot to another planet, with a little help from Star Drive of course. Here you see an exterior view of the Icarus with particles of space dust whizzing by set to a background of Apogee eclipsing the Sun ▼



I=Inventory, J=Jet

the planet Apogee. Your initial firepower consists of gas plasma lasers which aren't terribly accurate. Later on you can acquire much better weaponry ranging from Fire and Flee Missiles to the awesome destructive power of the Neutron Bomb which is required at the climax of the game.

Additional hardware such as this can be purchased in the refuelling depots deep below the planet's surface in the vast tunnel networks. These appear on many of the worlds in this galaxy.

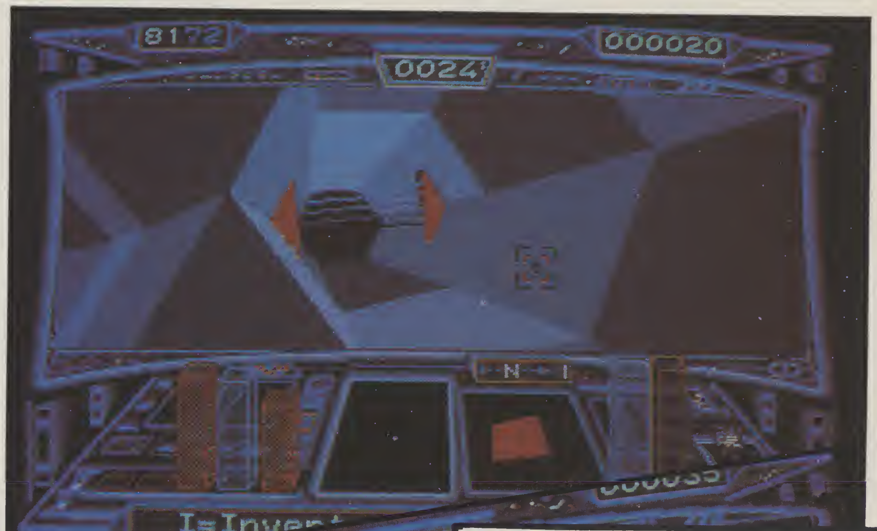
Once you've done everything you need to on the planet you can leave its atmosphere and head for another world with new challenges and rewards. Don't expect an easy time in space, however, as the space lanes are riddled with pirates and marauders hell-bent on pillaging your vessel. Rewards can be gained by blasting pirates and tractor-beaming their cargo into your ship. The loot can then be used to barter with the natives at your next port of call.

There are over 15 different planets and moons to visit, each with its own indigenous lifeforms and surface landmarks. The really nice aspect of this part of the game is the sense of freedom you have, in that, from the word go, you can travel anywhere in the Galaxy. This to my mind really makes the game work.

Both the graphics and the sound are of high quality. Particularly noteworthy are the sound effects which are not only effective but are useful as you can audibly identify objects such as ships. My only criticism is that the graphics are a little bit slow on some occasions. This is understandable when you think about the amount of data involved.

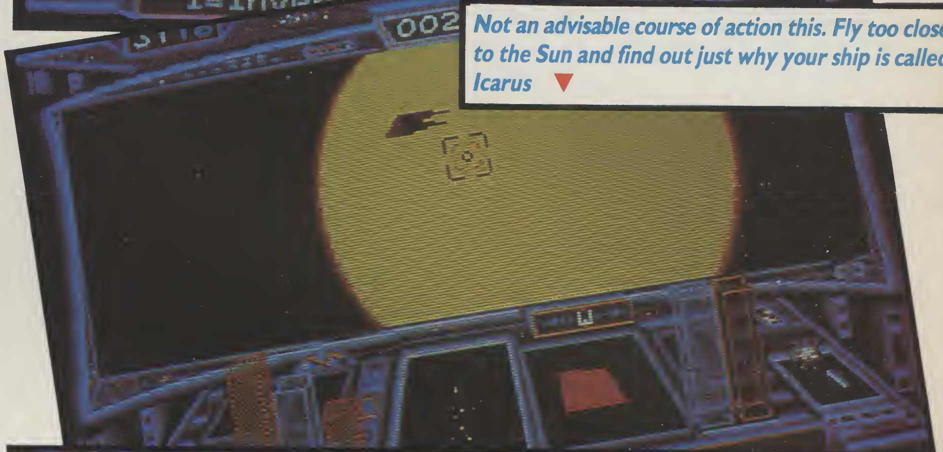
In short, *Starglider 2* is probably the best space cockpit game you're likeley to see for a good long while. *YA*

Having safely landed on Broadway, one of the moons of Millway, we now dive into the depths of the underground tunnel network where 'acquired' goods can be exchanged to fuel or weaponry by members of the resistance. Just ahead is a force field that needs to be taken out ▼

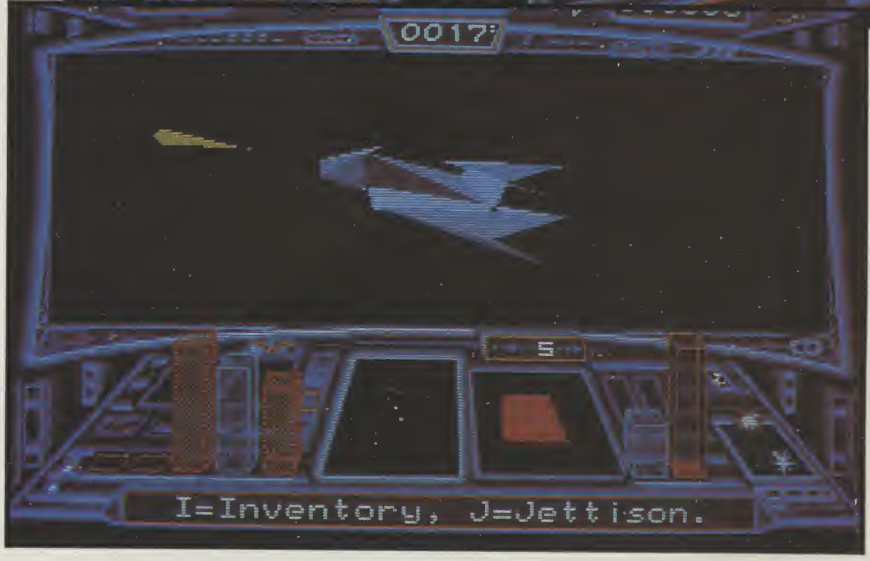


I=Inven

Not an advisable course of action this. Fly too close to the Sun and find out just why your ship is called Icarus ▼

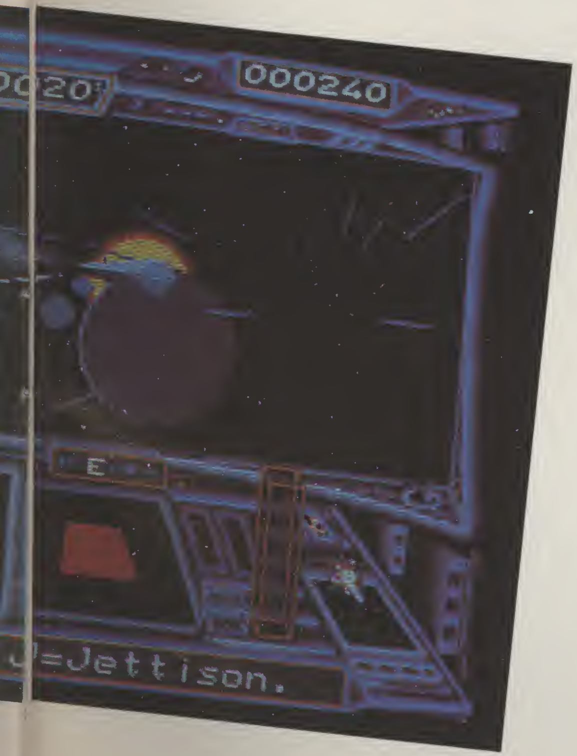


With the cursor keys just about every possible viewing angle is available from both inside and outside the cockpit. Here we see Icarus squeezing off a few rounds above the surface of Broadway ▶



I=Inventory, J=Jettison.

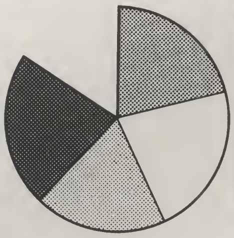
oot
ive
us
a
▼



J=Jettison.

STARGLIDER 2

Title: **Starglider 2**
Supplier: **Rainbird Software**
Wellington House
Upper St Martins Lane
London WC2H 9DL
Price: **£24.95**



▩ Graphics: **21**
☐ Sound: **22**
▨ Gameplay: **20**
▩ Value: **21**

Strictly for Beginners — Part 3

Newcomers start here. Allen Webb's series will turn you into an Amiga expert in no time flat

■ **I must say that it was most reassuring** to see some comments on the beginners' column in Issue 1 of *Your Amiga*.

Certainly the indications are that there are quite a few novices out there who are struggling to get to grips with the Amiga. This being the case, I hope that this series will help fill some holes. If any of you out there have any comments or constructive suggestions don't hesitate to drop *Your Amiga* a line.

This month, I plan to give a quick and dirty guide to graphics and the best software to look at. Although the manual gives a vague idea of the graphics capabilities of the Amiga, it really doesn't try very hard to do a decent job of it. In this article I hope to rectify the problem.

The first thing you need to realise is that the Amiga doesn't handle its screen memory in the same way as many of the 8-bit machines. Instead of using a block of memory which holds character values (as in the basic display on the 64), bit mapping is used throughout. This offers you the ease of the window and icon system but, as we will see later, at the cost of memory.

The Amiga offers two basic degrees of horizontal resolution. These are 320 and 640 pixels (or individual dots). The basic vertical resolution is 200 pixels but this can be increased to 400 pixels by use of interlace mode. Interlace mode uses a hardware fiddle to double the number of lines scanned. The first scan refreshes the even lines and the next scan refreshes the odd numbered lines. Since the last lines scanned decay during the next scan the display tends to be flickery, particularly if you use strongly contrasting colours such as black on white. The flicker can be eased by using a high-persistence monitor or, if you can afford it, a hardware add-on.

All graphics modes use these resolutions but provide varying numbers of colours and therefore make differing demands on memory. How colours are provided is a little involved but worth looking at in some detail.

The Amiga has 32 colour registers available which can be used to assign colours to dots on the screen. The problem comes when you attempt to store information specifying the assigned colours. Let us consider the most trivial situation of a two colour screen using colour registers 0 and 1. Consider the top left hand dot. This can have a colour register value of 0 or 1. The register value is therefore represented by a single bit which is either set or cleared. We can therefore envisage the screen to be a rectangular array of such bits. This array is called a "bit plane". If we want to use four colours, we would need two bits for each point. Two bits in binary offer the numbers 0 to 3 so we need two bit planes for each point:

For greater numbers of colours, we can use the relationship that the number of possible colours is equal to 2 raised to the power of the number of bit planes. 5 bit planes therefore allow 32 colours. (a five bit number represents the numbers between 0 to 32)The memory needed for any given graphics mode depends on the resolution and the number of colours required. Low resolution without interlace offers 320 by 200 points, or 64 000 bits. Pixels are stored at 8 per byte leading to a requirement of 8K per bit plane for low resolution mode. Table 2 shows the memory needs for the basic graphics modes:

Bit Plane 2	Bit Plane 1	Colour Register
0	0	0
0	1	1
1	0	2
1	1	3

You may notice that you cannot have more than 16 colours in high resolution mode. The other point of interest is that memory is soon gobbled up as you add more colours.

Table 1: Memory requirements of graphics modes (K)

	No of bit planes		Not interlaced		Interlaced
Number of Colours	High-res	Low-res	High-res	Low-res	High-res
2	1	8	16	16	32
4	2	16	32	32	64
8	3	24	48	48	96
16	4	32	64	64	128
32	5	40	—	80	—

Beyond these basic modes there are two additional modes of particular interest. The first is Hold and Modify (HAM) mode. This mode uses 5 or 6 bit planes to provide up to 4096 colours. This is achieved by a little trick. Four (or three) of the bit planes provide a pointer to one of 16 colour registers. The other two planes decide on the colour in a given dot. If the value of the relevant bits are zero in each of these planes, then the colour register pointed at by the other planes is used in its unmodified form. Otherwise, the colour register pointed at is used with one of its colour components modified. How this is done is shown in Table 2.

Inevitably there is a trade-off in this system. Since you can only alter one component each time you move a pixel, transitions between extreme colours can be tricky. If, for example you have a black pixel and you want to put a white pixel as close as possible to it. The least number of pixels over which this can occur is two and you end up with the sequence of colours: black, blue, cyan and white. This restriction isn't excessive and most software which uses HAM minimises undesirable effects. The use of 6 bit planes allows a full 4096 colours and 5 bit planes allows 240 colours.

The use of HAM allows superb quality pictures particularly if users have access to a digitiser linked to a video camera.

The final mode is "extra halfbrite" mode. In this mode the maximum 32 colour registers are supported along with their halfbrite values. Since the halfbrite values are linked to the registers, they change as the registers are altered. This system gives 64 "colour registers"

Table 2

Bit Pair	Effect
00	Colour in register unmodified
01	Register value with blue component taken from previous pixel (on left)
10	Register value with red component taken from previous pixel (on left)
11	Register value with green component taken from previous pixel (on left)

and requires six bit planes to provide the necessary pointer. This mode appears to be a recent addition and is not supported on all Amiga 1000 machines. In view of the relatively minor benefit accrued from the extra bit plane, I'm not convinced that extra halfbrite mode is that much use.

Table 3 shows the full range of colours available. Dashes indicate unsupported modes: Due to its open architecture the bit map memory is not always set up in the same area and can appear anywhere in the bottom 512K of memory. If you want to get at the memory you must locate the addresses of the Rastport (by using WINDOW (8) from BASIC) and the Intuition Window (by using WINDOW (7) from BASIC) data structures. The actual use of these data structures is a little beyond the scope of this article. If you want an idea how to get the addresses of the bit planes etc, howevere, take a look at the program called BitPlanes in the BasicDemos on your Extras disk.

Table 3

5	320x200	320x400	640x200	640x400	EHBx200	EHBx400	HAMx200	HAMx400
1	2	2	2	2	-	-	-	-
2	4	4	4	4	-	-	-	-
3	'8	8	8	8	-	-	-	-
4	16	16	16	16	-	-	-	-
5	32	32	32	-	-	-	240	240
6	-	-	-	-	64	64	4094	4096

In keeping with many other areas, the IFF format is used for graphics. This allows the interchange of pictures between different graphics packages. allowing the use of different facilities to create and refine pictures. The format of IFF files is fairly complex and the beginner should not get involved in trying to decode them. There are a number of sample programs on your Extras disk which allow the manipulation of graphics files. The three programs LoadACBM, LoadILBM-SaveACBM and Save ILBM allow you to mess about with Interleaved Bit Map Files (IFF) and Amiga Contiguous BitMaps. The latter format was developed to overcome the slow loading of IFF format.

Since it was bundled with the A500, most of you will have a copy of dPaint. Although it is long in the tooth, it gives an idea of what might be achieved on the Amiga. While I am at it, here are some pointers on its use.

■ The startup sequence doesn't initiate a printer driver. The disk does, however come with preferences. To set up your printer, type in "preferences" when prompted. Once you have set up your driver, leave preferences. You will be returned to the CLI command line.

■ To enter dPaint in low-resolution mode, simply type in "dpaint lo-res" and press enter.

■ To enter dPaint in medium-resolution mode, type in "dpaint med-res" and press enter.

■ To enter dPaint in high-resolution mode, simply type in "dpaint hi-res" and press enter.

It's a shame that the manual couldn't have told us that!

There are a number of quite sophisticated graphics packages available. What you choose really depends on what you want to do. Most offer HAM but this mode is not easy for creating pictures from scratch. It's much easier if you can do the basic work in a non-HAM mode first and then perform embellishments later.

Although it's not a definitive list, I can recommend that you look at the following packages:

■ Digipaint. This offers 4096 colour HAM only (in both interlace and non-interlace modes). It lacks some of the more sophisticated features now available but it operates quite quickly. Its shading and colour modifying options are effective and substantial. It will function happily with only 512K of RAM although some features are automatically disabled.

■ Photon Paint. This is a slick package (see the review in the August/September issue). It's certainly packed full of features but unfortunately limited to HAM (4096 colours). Again it will manage with 512K provided that you are happy with non-interlaced mode. If found the time of execution of some facilities rather slow.

■ Photolab. This is the best package I have seen to date. The big plus is that it allows you to operate in any of the graphics modes. This makes the development of pictures very simple. An added tweak is that you can hold an additional picture in memory to work on and it need not be in the same graphics mode as the other. The package also comes with a poster generating program and a colour processor. With the latter you can alter the tonal content of your picture, convert between modes or even remove bit planes. These options are very handy. Imagine that you have created a picture in HAM and you want to crunch it into less space for use in a game. You can convert it to 32 colour high-resolution mode and then, if required, remove bit planes. It is surprising how little difference it makes if you strip out a bit plane! The final delight is that this package comes with the 1.3 version printer drivers. These are fast and make life much more tolerable.

I hope that these comments have helped fill in some gaps in your appreciation of graphics on the Amiga. They are so powerful that it is a shame that their use isn't more fully explained in the manual. *YA*

One of C's greatest claims to fame is its ability to handle serious input/output problems effectively. This month Mark Burgess explains how to read and write data

FIRST WORDS IN C

Input/output on the Amiga can take a great variety of forms: simple line or character input from the keyboard (as is found found on all computers), reading and writing to files, mouse clicking or menu selection, or it might involve a combination of all of these, with a selection of windows and requesters thrown in for good measure. It would not do justice to the Amiga to attempt to write programs in a language which could not take full advantage of these facilities. In this issue, we begin by stepping through the basics of line and file input/output and show how to piece together programs which have complicated input/output requirements. The more difficult problems of dealing with Amiga windows require more preparation before they can be covered in this series.

Input/Output and the CLIs

Without any special programming to handle new windows, all of the input and output of a C program passes through a particular CLI window, namely the one which is used to run the program. All printed text appears in that window, and characters from the keyboard only reach the program if that window is the one currently selected by clicking with the mouse. Although AmigaDOS, the Amiga's file manager, and Intuition, the window manager, provide many ways of dealing with the Amiga's windows, here we concentrate on the very basics which can be performed using the standard C functions which exist on all computers possessing a C compiler.

For a great many applications, the simplest way to write text into an Amiga window, or to a disk file is to use the 'print formatted' functions which are provided by the standard C library. The *printf*() function has been used already in previous issues to print simple messages onto the screen and even to print out numbers. The companion functions *fprintf*() and *sprintf*() perform the same functions as *printf*, except that their output is directed towards files and strings respectively. The great advantage of using these functions is that, once you have learned how to use one of them, the others follow.

The general form of a *printf*() function call looks like this:

printf("...string..%1..%2..%3..",v1,v2,v3....);

v1,v2,v3... are variables or constant values of some kind. Remember that, in C, variables have both names and types. It is important to keep the type of a variable clear in mind when printing out its value because this affects the result in sometimes crucial ways. The string part of *printf*() is a message which you would like to be sent to the CLI window, just as in a normal BASIC PRINT statement. The difference is that, for every variable that you want to print out, it is necessary to put a marker in the string to show the exact place at which the result is to be placed within the string. This is done with the aid of 'conversion specifiers' or 'control sequences' which start with the per cent symbol (%) followed by some instructions. This may seem like extra work initially, but in fact it is much neater than trying to mix strings and variables in BASIC. Furthermore, there are some extras which C provides to make print formatting a pleasure rather than a chore. Some examples will help to illustrate how *printf*() works.

```
/****************************************/
/*                                      */
/* Listing 1 : printf demo              */
/*                                      */
/****************************************/
   #include <stdio.h>

   char *GLOBALSTRING = "..some text..";

/****************************************/
   main ()

   { int a = 33;
     char ch = 'x';
     float x = 2.367;

   printf ("This is just a message\n");
   printf ("The value of a is %d\n",a);
   printf (" a = %d \n ch = %c\n f = %f",a,ch,x);
   printf (" string -> %s\n",GLOBALSTRING);
   }
```

The program in listing 1 shows examples of how to print five different things: string messages, string variables, integer variables, floating point variables and character variables. Notice how each variable is positioned within the main string by placing a conversion specifier into the string at a particular place. The list of variables which follows that string echoes the list of conversion specifiers: the ordering of the list of variables is important and there is a different conversion specifier (%c,%d,%f,%s...) for each different type of variable. The output of the program in listing 1 is:

```
This is just a message
The value of a is 33
 a = 33
 ch = x
 f = 2.367000 string -> ..some text..
```

To illustrate the importance of type in the conversion, let's see what happens if the %d conversion specifiers in listing 1 are changed to %c specifiers. That is:

The %c conversion symbol means that *printf* now regards the value given to it as a character and so it will insert a character into the gap. Try recompiling listing 1 with these changes.

Notice that *printf*() does not print the decimal number 33 any longer, but instead prints the character whose ASCII value is 33 (the pling ! character). This gives a simple minded way of converting between ASCII characters and decimal values. However, beware of using this method too liberally: it does not make sense to write:

printf ("%s",a);

since there is no string which corresponds to an integer value, except for the garbage data which sits at the Amiga's memory address given by the decimal number.

The full list of conversion specifiers is given in table 1.

Try experimenting by printing out different values.

Special Characters

printf makes it simple to send control characters, or characters with unusual ASCII values to the screen by providing a 'backslash' notation. Control characters, including new line characters and carriage return characters, are invisible on the screen. They have special purposes usually to do with cursor movement, and often cannot be generated as characters by pressing any of the keys on the keyboard. In C, they may be typed into an ordinary string by typing a backslash character,' \ ', followed by some other character or a number. These

```
printf ("The value of a is %c\n",a);
printf (" a = %c \n ch = %c\n f = %f",a,ch,x);
```

```
This is just a message
The value of a is !
 a = !
 ch = x
 f = 2.367000 string -> ..some text..
```

***Table 1 : printf()* Conversion Specifiers**

%d - insert signed denary integer
%u - insert unsigned denary integer
%x - insert hexadecimal integer
%o - insert octal integer
%s - insert string
%c - insert single character
%f - insert fixed decimal floating point
%e - use scientific notation floating point
%g - use f or e, whichever is shorter

'backslash' sequences are listed in table 2. It is in fact possible to generate any number in this way, because if there isn't a special letter for the character you want, it is always possible to enter its ASCII value, in either octal (base 8) or hexadecimal.

Table 2: Control Characters for *printf()*

\ b backspace BS
\ f form feed FF (also clear screen)
\ n new line NL (like pressing return)
\ r carriage return CR (cursor to start of line)
\ t horizontal tab HT
\ v vertical tab
\ " double quote
\ ' single quote character '
\ \ backslash character

ddd character ddd where ddd is an ASCII code given in octal or base 8. (See Appendix C)
xdd character dd where dd is a hexadecimal number formed from two characters. (See Appendix C)

Formatting with Fields

Fields are vertical strips of the screen which separate data in some way. Columns of figures are said to live in fields, for instance. Each column is a new field on the screen. The *printf*() function permits the easy control of fields and column widths when printing on the screen. In addition it allows programmers to decide how numbers and strings will be 'justified' within fields.

Control of fields and justification is achieved by putting numbers in between the per cent sign and the characters which make up the conversion specifier in *printf*. For example, if you wanted an integer number to be printed into a field five characters wide, the conversion specifier would be %5d. This would mean that, however many digits were in the number, at least five character widths would be used to print it. Spaces would be used to fill in the extra width.

Normally numbers are pushed to the right of a field, so that the last digit is at the rightmost character position. The other possibility is to have the number 'left justified'. This is achieved by adding a - minus sign to the conversion specifier, for example: %-5d.

```
/***********************************/
/*                                 */
/* Multiplication Table            */
/*                                 */
/***********************************/

   #include <stdio.h>

   main ()

   { int i,j;

   for (i = 1; i <= 10; i++)
       {
       for (j = 1; j <= 10; j++)
           {
           printf ("%5d",i*j);
           }
       printf ("\n");
       }
   }
```

If you specify a field width which is too small, numbers will spill out and spoil your formatting. A simple example of numbers in rows and columns is a multiplication table. Listing 2 shows how to make one.

Floating point numbers have an extra feature over ordinary integers: you may choose the number of decimal places which are to be displayed. This is done by adding a 'point' into the conversion specifier, in addition to the other information. For example, %7.2f will print out a floating point number into a field of seven characters, to two decimal places, right justified. Again, if the number is too long to fit into the field, it will simply overflow.

You can also print strings in fields, by typing a number into the %s conversion specifier, as in %-20.3s. Here we have a minus sign which means 'left justify', a 20 which implies a field width of 20 characters and .3 which means that only three characters of the string are to be printed into the field. Try experimenting with C's print formatting.

Getting it in

Having mastered console output, using *printf*, the obvious next step is to explore its input counterpart *scanf*. The *scanf* function reads a line of characters from the keyboard and attempts to interpret it in some way: either as a string, or as an integer number or a floatingpoint number, or perhaps as several of these. A very versatile animal, but its great flexibility makes it slightly wild on occasions. Here we will show, first of all, how to tame *scanf* in the simplest possible way, to read in single numbers and strings. (see listing 3)

Unlike the input functions of other languages, *scanf* does not presume to impose its will upon how the user must type in numbers and characters, either by having to type into special field widths or by having to start floating point fractions with a zero instead of a decimal point, or whatever. Although *scanf* is very tolerant and quite intelligent when it comes to deciphering the user's input, but its tolerance only stretches so far. Where C differs significantly from languages such as Pascal or BASIC is in instances in which the input which it finds is totally incompatible with what it was told to expect.

For example, suppose you told *scanf* to read in an integer number and you typed in 'gobstopper', *scanf* would clearly be at a loss to find a sensible number to associate with that. Pascal would call a halt to the whole program and complain immediately, but *scanf* does not go to pieces over a little frivolity or stupidity on the part of the user (users frequently have one or other of those characteristics, after all), instead it will simply abandon its attempt to make sense of the input and can inform the program that the input make's no sense.

More than this, however, *scanf* will leave the nonsensical input in the input buffer, waiting to be read again, and won't change the variable which was to contain the data. It never discards nonsense simply because it cannot understand it. This very useful feature allows a program to go back and look at the input in a different way, or perhaps to simply discard it itself: at any rate, the input is in the program's hands, control is not snatched away by an angry input routine.

In listing 3 you will find a function called *skipgarb*() whose job is to empty an input line of characters. This is a very useful function for

```
Listing 3

/****************************************/
/*                                      */
/* CLI Input/Output Demo                */
/*                                      */
/****************************************/

   #include <stdio.h>


   main ()

   { int i = 0;
     float f = 0;
     char ch = '0';
     char string[20];

   string[0] = '\0';

   printf ("Current values are:\n\ni = %d\n",i);
   printf ("f = %f\n",f);
   printf ("ch = %c\n",ch);
   printf ("string = %s",string);

   printf ("\nNow please enter an integer number :");
   scanf("%d",&i);
   skipgarb();
   printf("The new value for i is %d\n",i);

   printf ("Enter a floating point (real) number: ");
   scanf("%f",&f);
   skipgarb();
   printf ("The new value for f is %f\n",f);

   printf ("Type in one characater: ");
   scanf("%c",&ch);
   skipgarb();
   printf ("The new value of ch is %c\n",ch);

   printf("An finally enter a string: ");
   scanf("%s",string);      /* better to write %19s */
   skipgarb();
   printf ("The new string is %s",string);
   }
/*************************************************/

   skipgarb() /* skip garbage on end of input line */
   {
   while (getchar() != '\n')
       {
       }
   }
```

dealing with bad input, since it ensures that, even if sensible answers were not read into a program by *scanf*, any 'garbage' which the user typed in, does not wait around to be read by the next *scanf*, messing that up too.

A few pointers

The *scanf* function works in almost exactly the opposite way to *printf*. You supply a conversion string, telling *scanf* how to interpret the stream of characters which is typed in, and also a set of variables to hold the resulting values. The syntax is:
n = scanf("%c %d", list of pointers to variables...)n is the number of variables which were sensibly matched (this tells you where the input started to go wrong, if at all). There is no need to include the 'n =' part if you do not want to know this information. The example listings simply ignore it.

Table 3 : Conversion Specifiers for *scanf*
%d int
%ld long int
%x (hexadecimal) int
%o (octal) int
%h short int
%f float
%lf long float / double
%e float
%le long float / double
%c char
%s string (pointer to char)

The conversion specifiers (almost the same as those for *printf*) are listed inside double quotes, giving the order of the types to be read in. Note carefully that if you type some text into these double quotes it will be ignored. The double quotes are not like those of BASIC's INPUT command. The only purpose of the double quotes in *scanf* is to establish a link between *printf* and *scanf*. Nothing will be printed out. Similarly any spaces in the string are ignored.

There is an important difference in the syntax of the variables between *scanf* and *printf*: look at the listings and notice the profusion of '&' characters in the variable lists, except for the string variable. These ampersand symbols mean 'give me the address of' and they are required because *scanf* wants to know where in the memory it is to store its values, not what the current values of those variables are. If you forget the ampersand from the *scanf* variable list, you could well end up crashing your Amiga, because *scanf* will try to use a variable value for the address of a place to store data. The compiler will normally spot this mistake and signal it as a warning.

Try typing in the example listing and playing around with the program by typing in all kinds of nonsense just to see how *scanf* interprets the input.

Files and Devices

Reading and writing to files is just as easy as reading and writing to the current console window. For so called 'text' filing, there are two new C functions called *fprintf*() and *fscanf*(). The only difference between these two and *printf* and *scanf* is that they take one extra parameter which identifies a file to be written to. Look at listings 4 and 5 for comparison with the old *printf* and *scanf* functions. The resemblance is more than just incidental: *fprintf* works in precisely the same way as *printf*, sending what would have appeared in your console window to a file instead. That file might be edited using a text editor. No compression is performed by *fprintf*.

Really there is no reason why file input/output should be any different from screen input/output: the only complication is that you now have to say where you want the data to be sent to, that is to identify a file by its filename. The process of identifying and locating a file is called 'opening' a file. So it is necessary, in fact, to open a file before it can be written to or read from. To open a file, you use the function *fopen*(). *fopen*() takes two string parameters: the first is the name of the file which you want to open, the second is a string which tells the file handler how the file is going to be used (whether it is to be opened for reading or for writing). *fopen*() returns a 'file pointer' if it successfully opens a file:
fp = *fopen*("name","r") This gives your program a single variable which identifies the file concerned. It is this file pointer which is used by *fprintf* and *fscanf*. If the file could not be opened *fopen* returns the value NULL for the pointer. It might happen, for instance, that the file is already in use, or that the file is protected in some way. This is something you should check for when opening files, because you cannot use a file which is not open.

The second of the two strings in *fopen* contains a single character: one of three: *r*, *w*, and *a*. These signify 'reading', 'writing' and 'appending'. They are three modes in which you can operate disk files. If you open a file for reading, you can only read from it. When it is opened, you are placed at the beginning of the file and as *fscanf* is used on that file, so you progress through the data until there are no more left.

If you open a file for writing, you can only write to the file. Again, you start off from the beginning and overwrite anything that was already in it. Appending a file also means writing data, but when a file is opened for appending, the starting position is the current end of the file and any new data is tacked onto the end, rather than overwriting existing data.

When a file is finished with, you have to close it, by calling *fclose(fp)* This frees the resources which the Amiga allocates to the file handling process and also frees the file to be used by another program. The example programs show how to code the opening and closing processes safely.

The Amiga treats all of its special devices as files and gives them so-called 'pseudo device names'. These include CON: and PRT:, the console window and the printer respectively. For all intents and purposes these devices are files. You may use their device names as though they were filenames, to gain control of a device. Listing 4 shows this in action with a separate console window used to display the output of the program. Note that the input still goes through the CLI window which called the program, so you have to click in that window to send keyboard data to the program.

Some devices cannot be opened in any old fashion. It does not make sense to open the printer device for reading, for instance, any more than it makes sense to print out data on the keyboard.

```
Listing 4

/****************************************/
/*                                      */
/* Console Window Input/Output Demo     */
/*                                      */
/****************************************/

   #include <stdio.h>

   #define code 0

/****************************************/

   main ()

   { int i = 0;
     float f = 0;
     char ch = '0';
     char string[20];
     FILE *fp;

   string[0] = '\0';     /* Init string */

   if ((fp = fopen("CON:100/100/400/100/OUTPUT","w")) == NULL)
      {
      printf ("Can't open window\n");
      return(code);
      }

   fprintf (fp,"Current values are:\n\ni = %d\n",i);
   fprintf (fp,"f = %f\n",f);
   fprintf (fp,"ch = %c\n",ch);
   fprintf (fp,"string = %s",string);

   fprintf (fp,"\nNow please enter an integer number :");
   scanf("%d",&i);
   skipgarb();
   fprintf(fp,"The new value for i is %d\n",i);

   fprintf (fp,"Enter a floating point (real) number: ");
   scanf("%f",&f);
   skipgarb();
   fprintf (fp,"The new value for f is %f\n",f);

   printf ("Type in one characater: ");
   scanf("%c",&ch);
   skipgarb();
   fprintf (fp,"The new value of ch is %c\n",ch);

   printf("An finally enter a string: ");
   scanf("%19s",string);
   skipgarb();
   fprintf (fp,"The new string is %s",string);

   printf("Press RETURN to close window\n");
   skipgarb();
   fclose(fp);
   }

/*****************************************************/

   skipgarb() /* skip garbage on end of input line */

   {
   while (getchar() != '\n')
      {
      }
   }
```

```
Listing 5

/****************************************/
/*                                      */
/* File Input/Output Demo               */
/*                                      */
/****************************************/

   #include <stdio.h>

   #define code 0

/****************************************/

   main ()

   { int i = 0;
     float f = 0;
     char ch = '0';
     char string[20];
     FILE *fin,*fout;

   if ((fout = fopen("output","w")) == NULL)

      printf ("Can't open output file\n");
      exit (code);
      }

   if ((fin = fopen("input","r")) == NULL)
      {
      printf ("Can't open input file\n");
      exit (code);
      }

   fprintf (fout,"Current values are:\n\ni = %d\n",i);
   fprintf (fout,"f = %f\n",f);
   fprintf (fout,"ch = %c\n",ch);
   fprintf (fout,"string = %s",string);

   fprintf (fout,"\nNow please enter an integer number :");
   fscanf(fin,"%d",&i);
   fprintf(fout,"The new value for i is %d\n",i);

   fprintf (fout,"Enter a floating point (real) number: ");
   fscanf(fin,"%f",&f);
   fprintf (fout,"The new value for f is %f\n",f);

   fprintf (fout,"Type in one characater: ");
   fscanf(fin,"%c",&ch);
   fprintf (fout,"The new value of ch is %c\n",ch);

   fprintf(fout,"An finally enter a string: ");
   fscanf(fin,"%19s",string);
   fprintf (fout,"The new string is %s",string);

   printf("Output written to file\n");

   fclose(fin);
   fclose(fout);
   }

Test Input File (create this using Ed to test listing 5)

12
2.5dThis is a string

E^@Output File

Current values are:

i = 0
f = 0.000000
ch = 0
string =
Now please enter an integer number :The new value for i is 12
Enter  a  floating  point  (real)  number: The new value for f is
2.500000
Type in one characater: The new value of ch is d
An finally enter a string: The new string is This
```

```
char string[30];
char filename[30];
static char dir[20] = "UserDirectory";

printf ("Enter filename: ");
scanf ("%s",filename);
skipgarb();

sprintf(string,"MyDisk:%s/%s",dir,filename);

fp = fopen (string,"r");
```

End of File

If a program continually reads from a file, eventually it will exhaust its contents. Once this happens, *scanf* cannot read sensible data from the file and it will respond as though it has no sensible input. To avoid the embarrassment of finding this out the hard way, there is a function called *feof*() meaning ‘end of file’ which returns a true/false answer to the question ‘have we reached the end of the file?’. You might use it in the following way:

if (feof(fp)) { printf (“End of file !”); }*fp* is a file pointer, obtained from a call to *fopen*.

Scanning and Printing Strings

With *scanf* and *printf* already under your belt, a further bonus is at hand. The formatting technology used by *scanf* and *printf* is also available for dealing with streams of characters in the form of strings. Two more commands called *sscanf* and *sprintf* perform the field-formatting/deciphering job, on arbitrary string variables. The syntax of these commands is very close to that of *fprintf* and *fscanf*, except that the file pointer variable is replaced by a string pointer, or the name of an array of characters. The general form is:

sprintf (sptr,’’..%c..%d..’’, variable list); n = sscanf
(sptr,’’..%c..%d..’’, variable pointers);

All strings are essentially arrays of characters and the easiest way to define a string in C is to declare an array of characters. The name of the array acts as a pointer to the string it contains.

As an example of how you might use this feature, consider putting together the filename of a file which lies in some default directory, defined by your program. You might have a string which contains the name of the directory on a disk with a volume name “MyDisk”. To get the full filename, you would need to amalgamate the volume name,directory name and filename to give a full path specification.

The possibilities for using the functions outlined here are vast. As an overview, we have covered the essential elements of text input/output in C. This is the kind of input/output which you would give to the console window, the printer and perhaps to disk files. C also provides more efficient filing functions for dealing with compressed data, though these are more advanced and require a knowledge of pointers and memory management, which we will cover in the next issue. *YA*

CLI
CLI
CLI
CLI
CLI — One Step At a Time

Make your own bootable disks with this third article on the Amiga Command Line Interface

By Burghard-Henry Lehmann

■ **CLI, the Command Line Interface,** lets you run programs directly. The most straightforward way to do this is by simply entering the name of the program you want to run. For example, if you've started CLI from your Workbench disk and enter *clock* this will open the Workbench Clock. Notice that the clock runs now from CLI and not from Workbench. To return to CLI close the clock by clicking the closing gadget on the left hand side of the clock window.

If *cd* is pointing at another disk than the Workbench disk you'll have to enter:

copy of a500 wb 1.2.:clock or *sys:clock*

If the program you want to run is sorted under a sub-directory you'll have to show CLI the path it has to take to find the program. For example, if you want to run Notepad from CLI enter: *utilities/notepad* (/ signifies a file or directory sorted under another directory), because the notepad program is sorted under the sub-directory *Utilities*.

By the way, if the name of a program consists of several words with spaces in between, you'll have to put the whole name in quotes. For example, *my program* would run a program called 'my program'.

Run

Since the Amiga is a multitasking machine you can run as many CLI processes as you like (and there is enough memory for). The CLI command *run* starts a new CLI process and executes whatever command comes after *run* on the old process. To understand this, enter *run clock*. First the new process will announce itself with *[CLI 2]*. Then the clock will be started as before. But now, even though the clock is running, you can return to CLI and enter whatever commands you like, including a command to run another program.

Run also works with other CLI commands, even though most of the time you'll want to use it to run a program while still being able to use CLI.

Newcli

You can also open an indefinite number of CLI windows and run a different task on each one. To open a new CLI window, enter newcli.

You might have noticed that a CLI window has no closing gadget. If you want to close a CLI window — any CLI window - enter *endcli* in the CLI window you want to close. In the last article I've talked about the difference between a system disk (*sys:*) and any other disk. A system disk can be used to start the Amiga of or "boot" it. Because of this it is also called a "bootable disk". The Workbench disk is of course a system disk.

You may also start the Amiga from CLI instead of Workbench. This has the advantage that CLI uses much less disk space than Workbench. Let's now format a disk and make it into a bootable disk. (For the following I assume that you've copied CLI to the Ram disk as I've explained in the first article of this series. This is important if you've only got the inbuild disk drive. Otherwise *format* and *install* will not treat the disk you want to be formated and made bootable.

First we format or initiate a new disk by inserting the new disk into the inbuilt drive and entering: *format drive df0: name NewDisk* (you may of course use any name you like). We have now formatted a disk we call *NewDisk*. Next, with *NewDisk* still in the internal drive, enter *install df0:*. Now *NewDisk* has been made into a bootable disk. Any disk can be made into a "bootable disk" with the *Install* command.

If you now reset the Amiga and start it off with *NewDisk*, you won't be asked to insert the Workbench disk as is normally the case, but the Amiga is started off on CLI. That is, you'll get a CLI prompt as you would get when running CLI from Workbench.

The only problem is that none of the CLI commands you enter will be recognized. Instead you'll get the error message *Unknown command* every time. This is because none of the CLI commands are on *NewDisk*. In order to be able to give CLI commands we have to copy the CLI commands you want to give first onto *NewDisk*.

Makedir

If you look at the directory of the Workbench disk you'll find that all the CLI commands are under a subdirectory called *c* — remember, in the first article we copied the CLI commands into the Ram disk by entering *copy c to ram:*. We now want to install a c directory on to

NewDisk. To do this we use the CLI command *Makedir* which creates a subdirectory on any disk.

Enter *makedir newdisk:c* if you now list the directory of *NewDisk* you'll see that indeed a subdirectory called *c* has been installed. To copy the CLI commands into that directory, enter *copy ram: to newdisk:c*.

I assume that you've copied all the CLI commands into the Ram disk as explained in the first article. At this point I'd like to give all of you, who, like me, have only got the one inbuilt drive, a good piece of advice: Whenever you want to copy a large file from one disk to another, copy the file first into the Ram disk and then copy it from there into the target disk. This saves you an awful lot of time changing disks!

As you might have noticed earlier on, as you've tried *NewDisk* which we made bootable, the font which *NewDisk* gives you on the screen is larger than usual. This is because when starting off on *NewDisk* the Amiga lacks certain information. This information is contained in a file on the Workbench disk, called *system-configuration*. You can find this file under the *devs* subdirectory.

To get the same configuration as on the Workbench disk on the disk we've made bootable, it is a good idea to first install a directory called *devs* on *NewDisk* and then copy the file *system-configuration* into that directory. In the next article I will expand on the theme of the bootable disk by showing how you can write your own startup-sequence which will automatically run a program and other things. This article I like to close with some more useful CLI commands.

Diskdoctor

Amiga disks have the habit of becoming quite easily corrupted, especially if you change a disk when the green drive light is still on! The result is a so-called hard disk error which the DOS will very quickly complain about.

To mend such a faulty disk, Workbench 1.2 includes a valuable facility, called *Diskdoctor*. Enter *diskdoctor df0:* and CLI will prompt you to put the faulty disk into the inbuild drive and press return. Then the good doctor goes through all the sectors of the faulty disk and tells you which sector contains a hard error. This is patched up, but the files on those faulty sectors are lost nevertheless. All *diskdoctor* allows you, is to save the non-corrupted files onto another disk and the reformat the faulty disk.

Diskcopy

CLI on Workbench 1.2 also lets you copy a whole disk. But this command is only really useful if you've got two or more drives or if you have copied CLI onto the Ram disk. On the whole you are much better off doing the job from Workbench, as described in the Amiga Manual.

Nevertheless, if you want to use this command, *diskcopy from df0: to df1:* will copy a disk in drive 0 to the disk in drive 1. **YA**

Rename

This allows you to rename a file or directory. *rename Our File to My File* will rename a file called *Our File* to *My File*. Please note, as always, if the file is under a subdirectory, you have to describe the correct path to the file in the instruction!

Relabel

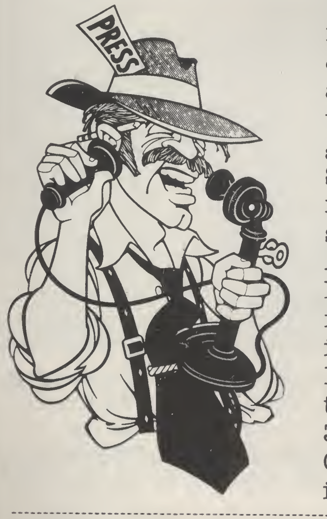
This lets you change the name of a disk — *relabel df0: NewDisk* will rename the disk in the inbuilt drive to *NewDisk*.

Fault

This is useful if a CLI command you've given has resulted in an error code, like *error 121*. In this case enter: *fault 121* and you'll get the answer: *Fault 121: file is not an object module* You'll get this error report whenever you've attempted to run a file which cannot be run, such as a textfile.

*** STOP PRESS *** STOP PRESS *** STOP PRESS ***

. . .GEORGE THOMPSON SERVICES advise all Amiga Owners that they have available the complete TBAG and AMUSE Public Domain Libraries STOP plus the latest FISH Disks fedexed in from the USA for the Commodore Show STOP All PD disks are £3.00 each or £30.00 for eleven inc. STOP For more information please refer to the new 'MEGA P.D. LIBRARY PACK' STOP Two disks for £4.99 STOP This includes a £5.00 Voucher redeemable against the first purchase of 10 or more disks STOP Regards Ed . . . END



"I couldn't have put it better myself Lou. Just one point to add! These guys are the originators of good PD on quality Sony disks. You may see libraries such as TBAG and AMUSE advertised later by others, but remember, you saw it here first! And Lou did you know they'll be on stand 32 at the Commodore Show in November?"

GEORGE THOMPSON SERVICES, specialists in good quality innovative software at competitive prices, can be contacted at DIPPEN, BRODICK, ARRAN, SCOTLAND KA27 8RN
TEL: (077 082) 234.

It's easy to complain about an advertisement. Once you know how.

One of the ways we keep a check on the advertising that appears in the press, on posters and in the cinema is by responding to consumers' complaints.

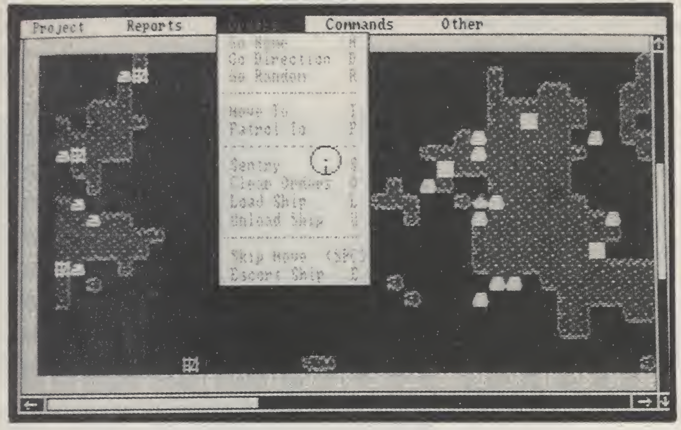
Any complaint sent to us is considered carefully and, if there's a case to answer, a full investigation is made.

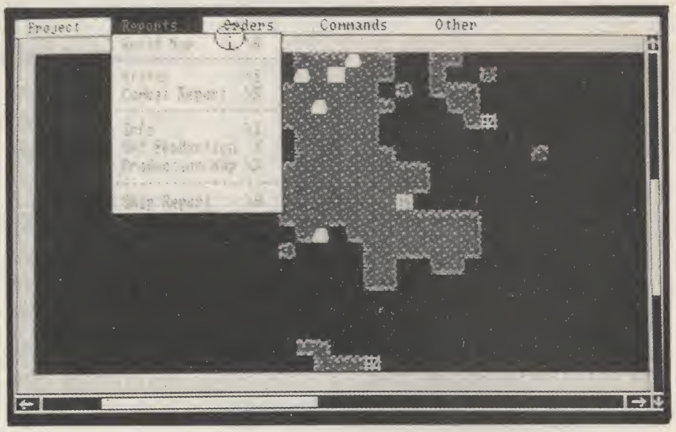
If you think you've got good reason to complain about an advertisement, send off for a copy of our free leaflet.

It will tell you all you need to know to help us process your complaint as quickly as possible.

The Advertising Standards Authority.
If an advertisement is wrong, we're here to put it right.
ASA Ltd, Dept 1 Brook House,
Torrington Place, London WC1E 7HN

This space is donated in the interests of high standards of advertising.





Fin Fahey finds himself satisfied with nothing less than total global conquest. But can he stop playing Empire first?

Empire

■ Character graphics on the Amiga? At first sight, this is one of the most unprepossessing games I've seen on the Amiga, short of the appalling *Around the World in 80 Days*. Let me say right away that surface appearances are deceptive. Although it may look like a survivor from the Bad Old Spectrum Days, *Empire* had my midnight oil bills soaring through the roof before I was finished reviewing it.

Empire, like many of the most addictive strategy games, is a production-based wargame. The action takes place on a very large map — you can choose from one of six or design your own — and the accent is on naval power. The seas must be dominated, or transport ships, used to carry armies, simply will not get through. The naval units form a hierarchy from destroyers to battleships, and take an increasingly large number of game turns to produce as you go up the ladders. The crucial game squares are cities, where all this production takes place, and since you only start with one, there's a mad scramble to get your production on a good footing right at the start.

The basic game rules are astonishingly simple. Any piece next to a square can attack it simply by trying to move there, and then it either destroys its opponent or is destroyed depending on the balance of power and the levels of damage of the two pieces. But around this simple idea, Interstel has spun a game with many facets which also has a strange air of realism.

First off, there's the adventure element. The game starts out with you only knowing about nine squares of the planet you're contesting. As you send out ships and air patrols, more and more becomes apparent, but there's always that urge to find out what's just around the corner. Then there's the range of game features. You can make life easy for yourself as time goes by and you acquire a large number of units. This could be hard, since on each turn each must have some sort of orders, even if they are to 'pass'. Each unit, however can be either left on sentry duty or told to go somewhere, in which case they don't bother you for a while.

That's conventional enough, but the Patrol option offers you something I haven't seen before. With this, you can order a piece to move regularly between two points on the map, and not bother you until it runs into something. Why doesn't SSI include this kind of thing? I also liked the option you have of naming your naval units as they're produced. It's a drag playing games where your new battlewagon, pride of the fleet, is called something like B9. With *Empire* you can name your entire navy after the WWI German High Seas Fleet, or whatever. The annoying thing is, though, that the program has a habit of taking the naming out of your hands from time to time, so that, after naming my ships consistently after the WW2 Royal Navy, the computer foisted a *Bismarck* on me. Beastly bad show, I say.

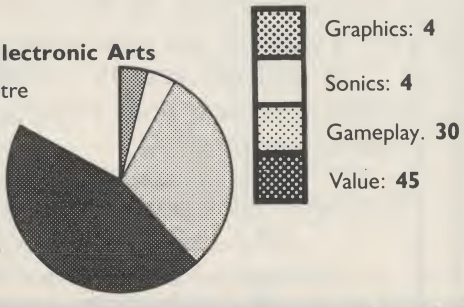
Of course, features or no features, this game would be worthless if it didn't offer any decent opposition to your warlike schemes. Leaving aside the PBM(Play By Mail) and human opponent options, this is the first strategy game I've played in ages that really gives you a run for its money. Up to two other opponents can be fought, either or both of which can be computer controlled.

The machine player managed to avoid most of the usual computer wargame flaws. After all, computers are idiotic machines, and programming them to play something several orders of magnitude more complex than chess is totally daunting. It can't be done — yet — but in the mean time Interstel has managed to avoid two major problems. First, the machine isn't wholly predictable — it doesn't commit forces piecemeal, but turns up in strength in unexpected places. Second, the accent is on strength. The important thing in wargames is not to spread yourself thin, but to always concentrate forces as much as possible. This *Empire* does, which completely confounded my expectations of the opposition. Don't worry, I'll still beat it — it's just going to take me that bit more lost sleep.

I didn't expect to like this game, but quite honestly it's the best war game I've seen in months. The designers could have based the map on hexes not squares, and improved the mouse control, but these are minor quibbles. I'd rather have one of these than a thousand of the good-looking but, in my opinion, completely vacuous, releases from Cinemaware. So there. *YA*

EMPIRE

Title: **Empire**
Supplier: **Interstel/Electronic Arts**
Langley Business Centre
11-49 Station Road
Langley
Berks SL3 7YN
Tel: **(0753) 49442**
Price: **£24.95**



Now you can afford perfect data storage

(Available in boxes of ten: 5.25" SS/DD at £7.95; 5.25" DS/DD at £9.95; 5.25" DS/HD at £15.95; 3.5" DS at £17.95; all inclusive of VAT)

The all new range of Commodore Diskettes

Don't risk losing your valuable data. The superb quality offered by Commodore Diskettes ensures perfect data storage, time after time at a price you can afford.

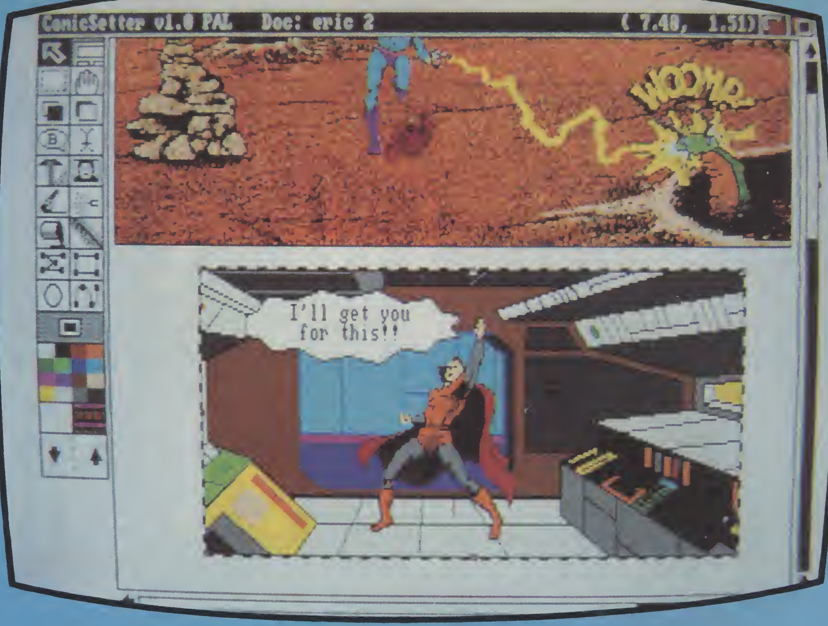
Available now from your local computer dealer

Commodore

The Original Diskette - Higher Quality, Lower Price

Trade enquiries only to SJB Disks Ltd., the officially appointed sole U.K. distributor.

Aaaargh! Zap! Pow! Will you look at the size of that thing! Alan Batchelor wrestles with the Amiga approach to comic art



Comic Setter

Faster than a speeding bullet! More powerful than a locomotive! Able to leap tall buildings in a single bound! Is it a bird? Is it a plane? No, it's Comic Setter. This interesting package comes in a colourful box, decorated with 1940s type superheroes telling you that you, too, can create "killer comic fonts, word balloons and boffo birthday cards." All in luscious color. what more could an aspiring Stan Lee ask for?

On opening the box I was faced with two disks, a short science fiction comic strip created with the software itself, and a 104 page manual. The first problem appeared too. You need at least 1 megabyte of memory to run this program.

Once Comic Setter is loaded you have to create a page. The menu system used makes this a simple task, and all it means is that you've got a blank white page in front of you, ready to put the panels in. Using the panels icon, click your mouse button once where you want the top left corner to be, and again at the bottom right. Lo! There's a frame all set to draw in — you can, should you wish, move the panel around.

Next, it's time to choose a background. There are quite a few supplied on the clip art disk, ranging from apartments and offices to lunar landscapes. All the backgrounds look as though they've come straight from a 1960's Marvel comic. They're simple yet well drawn, and one or two have been done at the sort of angle they used to use in the Batman TV series.

Next, my personal favourite - choosing characters to put into the picture. Again, there's a wide choice on the disk, and not only can you use pre-generated figures, but you can build your own. Holy construction kit! Strangely enough, apart from the spaceman figure, all the men and women are in swimming costumes. I wonder why...

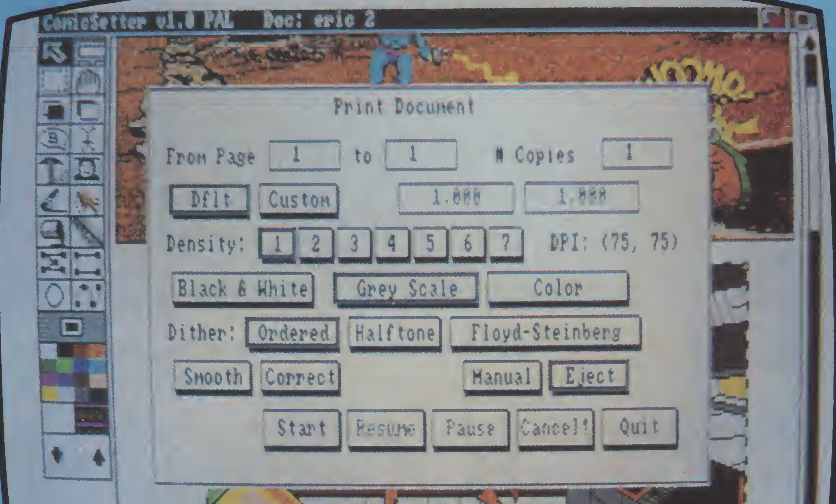
When you've chosen your figures and positioned them to your taste, you can add speech bubbles. Even here, you're not limited to one shape, as you're given speech, thought, and shout bubbles. The latter are the ones with jagged edges — you can decide on the number and size of spikes, incidentally.

The program lets you use any of the 4096 colours available on your Amiga as your working colours, and by double clicking on the colour palette you get access to a control for altering hue, luminance and saturation. some interesting effects to be had here, I think.

If you want to draw your own stuff, you have to choose between bit-mapped graphics or structured drawings. The former uses graphics created with other art packages — the manual mentions dPaint II. Structured drawings, on the other hand, are drawn in your panels using the graphic tools in the program. The main tools are line, box, ellipse and bezier. The first three are self-explanatory, but bezier is a way of producing curves similar to those in graphs.

If you'd like text in your pictures, set the graphics tool to structured, select the text icon,and use the cross hair cursor in the same way that you did when creating your panels. There are three styles of text for you to play with,and a variety of fonts as well.

If you're like me, the editing facility will be well used. in this case, on selecting the edit command, you get a menu of various editing functions, with

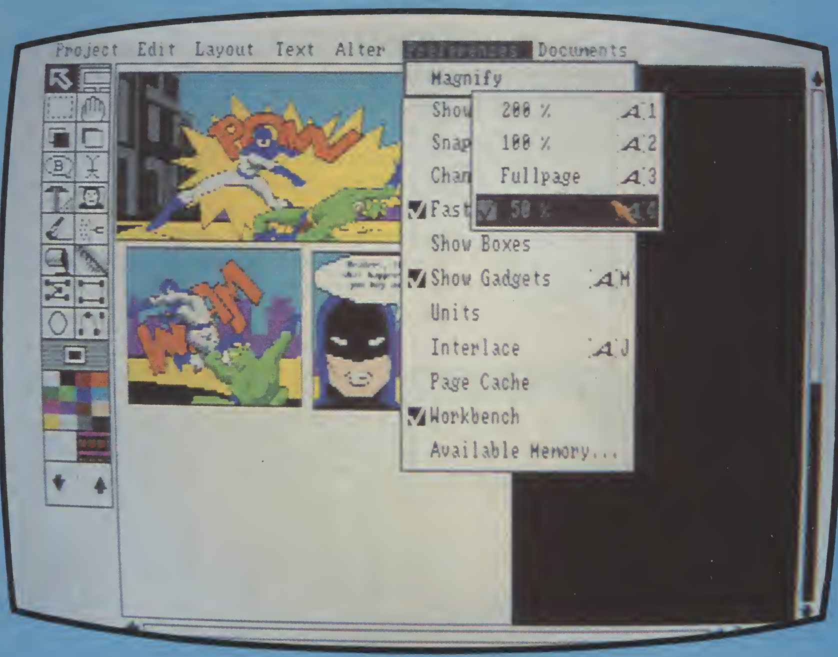

ComicSetter v1.0 PAL Doc: eric 2

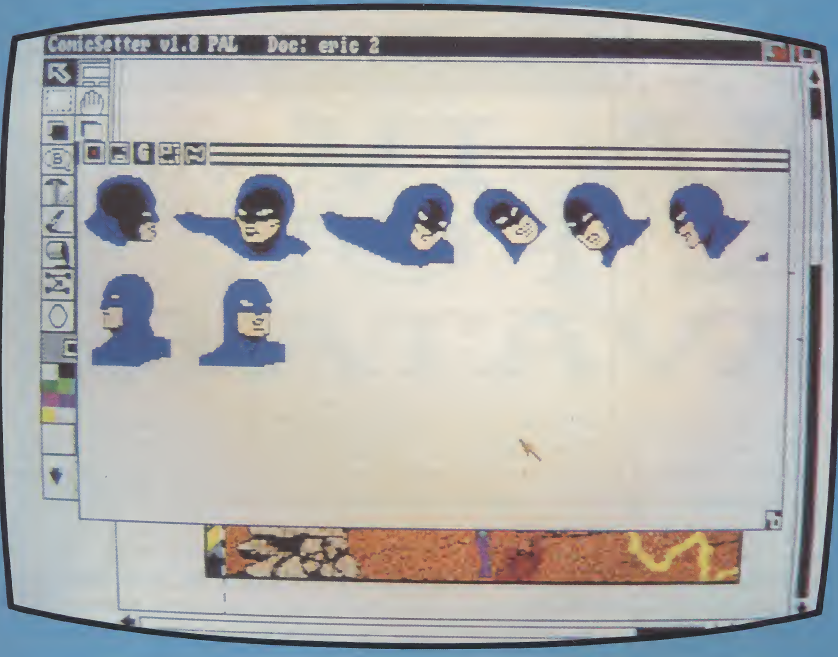
Print Document

From Page 1 to 1 # Copies 1

1.000 1.000

Density: 1 2 3 4 5 6 7 DPI: (75, 75)





the ones currently available highlighted.

For those perfectionists among you, there's the option of a grid system, whereby you can place the panels more precisely. Grid sizes can be changed both horizontally and vertically.

Where would a comic book be without its print run? Nowhere, that's where. Fortunately the printer options are extensive and should cover everyone's needs. There are three expansion sets available. First, science fiction. This is full of aliens in capes, rugged heroes and a nice line in mutant worms. Backgrounds are a little basic, being simply drawn planetscapes and futuristic cities, but if you put enough starships in, who's going to notice?

Second, we have superheroes. Yes, lots of little men running around in their underwear. The facial close-ups of the villain remind me of Alan Davis' current work on Excalibur. They really are that good. Backdrops are Manhattanesque since as everyone knows, this is where almost all the superguys live. Of the three, this is my favourite.

Last and possibly least, according to your taste, is the funny figures pack. This should have been the one I raved over. I loved Captain Carrot. I laughed at Bugs Bunny. This, however, leaves me cold. One or two of the backgrounds weren't too bad, notably the forest snowscene and alleyway, but I can't take to the characters themselves. They are too simple, and their expressions lack something I can't quite put my finger on. The saving grace on this disk is the choice of props available. Swords, catapults, anvils, magnets, you name it. This is where you can have a laugh. All the heroes, monsters, cute little mice and spaceships are interchangeable. The thing to do is have a big hunky guy in a cape being flattened by a duck with a sledgehammer - the possibilities are almost endless.

Thanks to the manual, which is well thought out and logical, it shouldn't really take too long to master the seemingly hundreds of things you have to learn. All you need is a little perseverance.

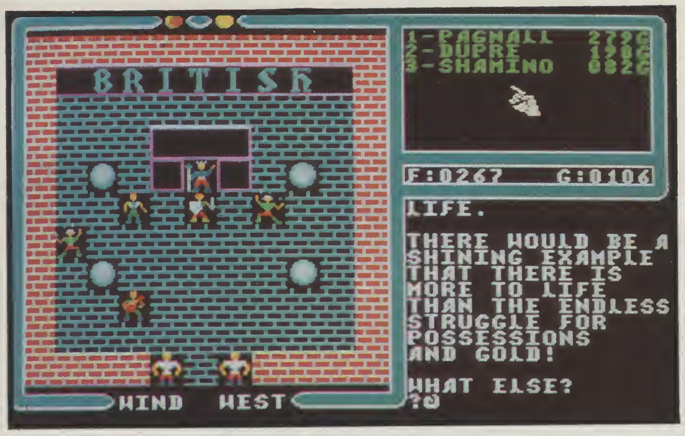
Comic Setter is a must. The price may be a bit steep,but you'll be kept amused for hours. So, all you comic fans especially, let your imaginations run riot. Finally, some last pointers on using Comic Setter properly. Be pure. Be vigilant. Behave. *YA*

COMIC SETTER

Title: **Comic Setter**
Publisher: **Gold Disk**
Streetsville,
Mississauga,
Ontario,
Canada L5M 2C2
P.O. Box 789,
Price: TBA

ULTIMA IV — QUEST FOR THE AVATAR

Can Gordon Hamlett change all his vices into the eight virtues necessary to save the land of Britannia from evil? Read his review of this latest roleplaying game.



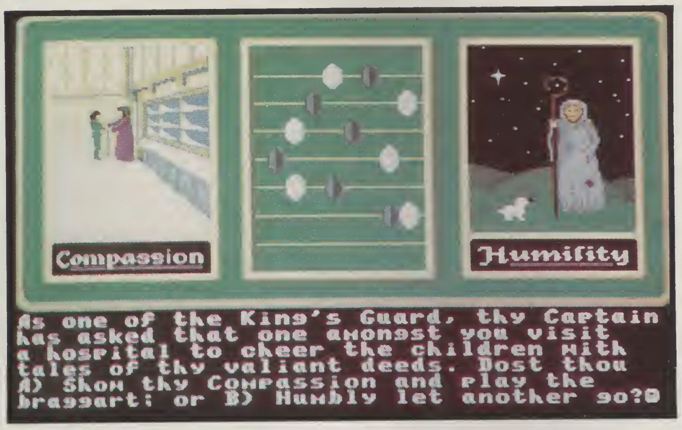
```
BRITISH
1-PAGNALL   272G
2-DUPRE     178G
3-SHAMINO   082G
F:0267      G:0106
LIFE.

THERE WOULD BE A
SHINING EXAMPLE
THAT THERE IS
MORE TO LIFE
THAN THE ENDLESS
STRUGGLE FOR
POSSESSIONS
AND GOLD!

WHAT ELSE?
?
WIND  WEST
```

Such were the forces released when Exodus was finally destroyed in *Ultima III*, that the land of Britannia underwent mighty upheavals. Mountains and islands appeared from under the oceans and entire land masses disappeared. Once again, Lord British managed to bring peace to the new land and Britannia prospered. With this new found wealth, came a desire for self improvement, searching out the path of the eight virtues by means of meditation and other divers methods.

Your appearance in Britannia, although somewhat baffling to yourself, had long been predicted by those who studied these things. Out for a walk one day, you discover an ankh on a chain and a couple of books. Although you don't realise it, you have already started on the path towards Avatarhood. You walk into a mediaeval fair although you have no idea whence it came. Your feet seem to drag you towards an old gipsy caravan where an old crone insists on telling your fortune. She asks you a series of questions based on various moral dilemmas to which there is no right or wrong answer. According to your replies, so your character is determined for the quest ahead. No random rolling of the ivories for you.

As your journey starts, so you are on your own. There are eight different professions in Britannia: Ranger; Paladin; Tinker; Druid; Fighter; Mage and Shepherd. You will have to recruit seven other characters (one from each class) to your party as the game progresses. Not all of them will join you straight away though. To some, you will have to prove your worth first.

In order to achieve full Avatar status, you must first become a partial Avatar in each of the eight virtues. These are spirituality, compassion, honour, justice, sacrifice, valour, humility and honesty. Apart from doing good deeds like giving money to beggars and so on, you will also need to discover the appropriate mantra and stone for each virtue, then find and meditate in the requisite shrine. Only if you are good enough will you be granted a vision.



```
As one of the King's Guard, thy Captain
has asked that one amongst you visit
a hospital to cheer the children with
tales of thy valiant deeds. Dost thou
A) Show thy Compassion and play the
braggart: or B) Humbly let another go?
```

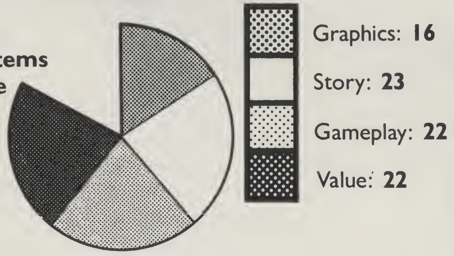
Magic also plays an important part of the game, both in and out of combat. Various characters have a limited spell casting ability but only mages are true masters of the art. Spells have to be pre-prepared and that involves not only having the correct ingredients, but also mixing them in the correct proportions. Six of the eight ingredients, blood moss, spider silk, black pearl, ginseng, garlic and sulphurous ash can be readily obtained at the local herb shop but the other two can be found only in the dead of night at specific locations.

Acquiring information is done by talking to each and every one of the hundreds of characters that you come across in the game. Initial enquiries can be made as to their name, health and occupation and from their answers, so you might discover the next key word to ask them about. Frequently this will involve going to find somebody else. An example is: 'Go to the pub in Jhelom and ask about sextants'. Not only will you have to keep copious notes, but you will also have to backtrack a lot across the countryside. As well as walking, other, faster modes of transport include horse, boat, balloon and moon gates. This latter method is a system of teleportation dependent on being in the right place at the correct conjunction of the two moons.

That apart though, *Ultima IV* scores over most of its other roleplaying rivals in its playability. The game system is entirely logical with no glaring anomalies such as are present in other rival products. With an estimated playing time of between 100 and 200 hours, *Ultima IV* cannot be recommended too highly. **YA**

Ultima IV

Title: **Ultima IV**
Authors: **Origin Systems**
Suppliers: **MicroProse**
2 Market Place
TETBURY
Glos GL8 8DA
Tel: 0666 54326
Price: **£24.95**



Graphics: **16**
Story: **23**
Gameplay: **22**
Value: **22**

commodore

FROM £375!

AMIGA

Amiga A500 complete, now only	£375
Amiga A500 with TV modulator	£399
Amiga A500 with 1900M high-res mono monitor	£465
Amiga A500M with A1084 high-res colour monitor	£615
Amiga B2000 with 1MB RAM, 880K 3.5" disk, mouse, software	£1095
Amiga B2000M As above, plus A1084 hi-res colour monitor	£1245
Amiga B2000 XTM As above, plus PC-XT bridge board & floppy	£1695
Amiga B2000 XTMHD As above plus A2090PC 20MB hard disk	£2145

Prices include 15% VAT, FREE delivery to your door, and 1 year manufacturer's warranty. Add £12 for next day delivery. All systems are tested before despatch. On-site maintenance option available.

PERIPHERALS

AMIGA

512K plug-in RAM/clock (A500)	£115
A1081 high-res colour monitor	£225
A1084 high-res colour monitor	£249
1900M high-res mono monitor	£99
A1010 3.5" external disk drive	£149
3.5" twin-port external drive	£115
3.5" internal drive (2000 only)	£99
Genlock PAL Rendale	£275
Genlock PAL Rendale Pro	£799
A2052 2MB internal RAM card	£375
Micron 2MB internal RAM card	£399
Supra 20 MB external hard disk	£675
A2092A 20MB internal hard disk with Amiga DOS controller	£599
A2092PC 20MB internal hard disk with MS DOS controller	£499
A2088 PC-XT bridge board with 5.25" internal disk drive	£475
Citizen 120D 120 cps, F/T, NLQ	£149
Star LC10 120 cps, F/T, NLQ	£215
Star LC10C as above, colour	£245
Canon PJ1080A colour, 80 dpi	£499
HP DeskJet 300 dpi inkjet, B/W	£795
HP PaintJet colour, 180 dpi	£995
Xerox 4020 colour, 240 dpi	£1095
Handy Scanner 240 dpi scanner	£299
DigiPic fast frame grabber	£225
Easyl A4 drawing tablet A500	£275
Easyl A4 drawing tablet A2000	£325
Cherry A3 digitizing tablet	£599

CREATIVITY

AMIGA

De Luxe Paint II (PAL)		£59.95
DigiPaint (PAL) 4096 colours		39.95
De Luxe Music (PAL)		59.95
Page Setter		79.95
City Desk		79.95
Excellence!		179.95
The Works	Analyze + Organize + Scribble, Integrated desktop	59.95
VizaWrite Desktop	High performance desktop publishing wordprocessor	69.95
ProWrite (V2.0)	Wordprocessing fully integrated with graphics	79.95
Word Perfect	Amiga version of the No.1 best selling wordprocessor	179.95
Publisher Plus	Enhanced version of the original desktop publisher	99.95
Professional Page	Includes WP, Desktop, colour separations, CAD	199.95
DeskTop Pack	VizaWrite Desktop + City Desk	149.95
Flow	The Idea Processor from New Horizons Software	49.95
De Luxe PhotoLab	Print & manipulate photo quality images	59.95
PixMate	Professional image enhancement straight from NASA!	44.95
Prism Plus	4096 HAM colour paint pack 1024 x 1024 (needs 1MB)	54.95
Express Paint	Overscan PAL + 64 colour Extra Half Brite, text merge	49.95
Photon Paint	HAM with overscan, special effects, enlarged fonts	59.95
TV Show	PAL Image presentation in IFF or HAM with overscan	69.95
Graphics Pack	Express Paint + PageFlipper + IntroCad	99.95
Calligrapher	Professional font editor, fonts up to 160 x 256 pixels	74.95
De Luxe Video II	Create all types of video presentations with ease	59.95
De Luxe Productions	Definitive high-res graphics animation package	129.95
Aegis VideoScape	Full 3D animations in all resolutions with overscan	124.95
Pro-Video CGI	Professional video titler with fonts, extra fonts available	159.95
Intro Cad	Entry level CAD package, printer or plotter output	49.95
X-Cad	Professional CAD system (needs 2MB)	399.95
Art of Chess	The best Amiga chess yet, packed with features	15.95

Superbase

PROFESSIONAL

If you can think of it, **SuperBase Professional** can do it! With its unique combination of incredibly simple data management, massive processing ability and high-level program language, **SuperBase Professional** is the only choice for your Amiga! And with text, sound and graphics management, plus relational data handling, **SuperBase Professional** is essential for both beginner and expert...

PRODUCTIVITY

AMIGA

SAM Basic		£89.95
A/C Basic		154.95
A/C Fortran		229.95
Lattice C 4.0		124.95
Lattice C Professional		299.95
MCC Pascal 2.0		69.95
64 Emulator	Use your C64 programs and files! Includes cable	49.95
PC Emulator	The Transformer (MS-DOS not supplied)	29.95
SuperBase Personal	Relational database power, without programming!	69.95
SuperBase Professional	With Forms Editor and DML program language	179.95
Acquisition (V1.3)	Relational database with programming language	174.95
Maxiplan 500	Ultimate Amiga spreadsheet, + text/graphics/speech	89.95
Maxiplan Plus	As Maxiplan 500 above, plus powerful Macro language	129.95
VIP Professional	1-2-3 compatible spreadsheet + database & graphs	99.95
Logistix	2048 x 1024 spreadsheet, database, graphs & calendar	99.95
Power Business Pack	SuperBase Personal + VizaWrite + Logistix	199.95
Businessware Accts 1	Invoicing, Sales Ledger, Stock Control, by Panmead	129.95
Businessware Accts 2	General Ledger, Purchase Ledger, by Panmead	129.95
MCC Assembler	Professional development system with linker & libraries	59.95
K-Seka Assembler	Efficient memory-resident assembler	29.95
MCC Shell	Enhanced CLI by the authors of AmigaDos	29.95
Ultra Dos Utilities	High speed disk backup, ideal for hard-disk users	34.95

■■■ IF YOU WANT IT TOMORROW... CALL US TODAY! ON 01-546-7256 ■■■

Prices are POST FREE & include VAT. Order by phone with your credit card, or send cheque/PO or your credit card number. Official orders welcome. We despatch same day by FIRST CLASS post. Please allow 5 days for delivery of hardware orders. Prices are quoted subject to availability. Ref. A58

Access

VISA

Calco Software

LAKESIDE HOUSE, KINGSTON HILL, SURREY, KT2 7QT. TEL 01-546-7256

NEW FROM TRILOGIC

NEW - AMIGA AUDIO DIGITISER

Superb performance, with adjustable sensitivity, and overload LED indicator. Supplied with connecting lead to personal stereos, radios, etc. Works with all existing commercial software, e.g. Prosound, Audiomaster, Datel Prosampler etc.

A500 DIGITISER, LEADS & INSTRUCTIONS ONLY **£24**.99

NB GENDER CHANGER REQUIRED FOR A1000 £2.98 EXTRA.

PRINTLINKS

USE YOUR C64 PRINTER WITH THE AMIGA

All Printlinks enable you to use your CBM Serial printer e.g. MPS801 with most home computers, e.g. AMICA, ATARI, IBM PC/AT/XT etc.

All make use of your redundant COMMODORE 64 or 128 as both buffer and interface and consist of an interface cartridge with integral software.

PRINTLINK 1

PARALLEL TO CBM SERIAL PRINTER CONVERTER

Use Printlink 1 with most computers and one CBM Serial Printer.

- Parallel Input — connect to the computer printer port.
- CBM Serial Output for Text Use on Commodore Serial Printers.
- 60k Printer Buffer — frees computer while documents are printed.
- Printer Buffer Commands — Pause, Repeat & Terminate etc.

ONLY £34.99

GET THE PICTURE? — IT'LL BE SHARPER, AND CLEARER WITH A TRILOGIC AMIGA TO TV RGB LEAD AND COULD SAVE YOU £££'s.

We have leads to connect all AMIGAS to your TV or colour monitor provided it has an RGB input socket. All leads give a much clearer picture than using the AMIGA MODULATOR, permit ALL 4096 colours to be displayed and include the audio lead (to give stereo with stereo tv's).

ORDER **AL 1** FOR TV's WITH 21 PIN EURO (SCART) SOCKET. FITS PHILIPS, FIDELITY, SONY, GRUNDIG, NORMENEDE, ETC. **ONLY £9**.99

ORDER **AL 2** FOR FERGUSON TV's WITH 7 OR 8 PIN DIN SOCKET. MODELS MC01 & MC05, ETC. **ONLY £9**.99

ORDER **AL 4** FOR HITACHI & GRANADA TV's WITH 7 PIN DIN SOCKET MODELS CPT1444, ETC. **ONLY £9**.99

SORRY. LEADS ARE NOT AVAILABLE FOR THE FOLLOWING MAKES: MICROVITEC, AKAI, SEISHO & PHILIPS CF1114 WITH 5 or 6 PIN TTL RGB SOCKETS.

LEADS ALSO AVAILABLE FOR ATARI ST RANGE.

PLEASE CONSULT US IF IN DOUBT.

ATTENTION 1901 MONITOR OWNERS.

WHY NOT HAVE YOUR 1901 MONITOR CONVERTED TO WORK WITH THE AMIGA?

Our conversion enables your 1901 to display all 4096 colours and gives a really excellent picture. We have converted hundreds of 1901's with 100% satisfaction. Price including the lead is ONLY £29.95 — for callers. Phone to arrange an appointment or we can collect, convert and return your monitor ALL WITHIN 4 DAYS for the all inclusive price of **£53.95**.

OTHER INEXPENSIVE AMIGA LEADS

	PART NO.	PRICE
2nd DRIVE SWITCHED LEAD — Disables external drive to permit programs to load.	ADE 2	**ONLY £9.99**
2nd DRIVE SWITCHER — Fits between drive connector and Amiga disk drive port. Can be used when power is on.	ADE 3	**ONLY £8.49**
JOYSTICK PORT EXTENDER — Fed up with groping around the rear of your Amiga — then you need this lead.	JPE 1	**ONLY £3.99**
MOUSE/JOYSTICK SWITCHER & PORT EXTENDER — Extends the port and has sockets for mouse and joystick with push-button switch for mouse or joystick selection.	DJA 1	**ONLY £9.99**
3 & 4 JOYSTICK ADAPTORS — Some games use 3 or 4 Joysticks — you'll need this lead to use more than 2 Joysticks.	JEL 1	**ONLY £7.99**
AMIGA 64 EMULATOR LEADS — Connects 1541 ETC to your AMIGA. There are several programs which need this lead.	AEL 1	**ONLY £4.99**
MODULATOR EXTENSION LEADS	MEL 1	**ONLY £8.99**
AMIGA PRINTER LEAD — Parallel Type 1.5M long	AMP 1	**ONLY £6.99**
AMIGA PRINTER LEAD — Parallel Type 3.0M long	AMP 3	**ONLY £9.99**

PLEASE STATE WHICH AMIGA YOU HAVE WHEN ORDERING.

HAVE YOU HEARD YOUR AMIGA'S AMAZING SOUND IN STEREO?

AMIGA MINI-AMP 1

This neat stereo amplifier connects directly to your AMIGA and boosts the audio output so that you can enjoy the AMIGA's amazing stereo sound in privacy, using 1 or 2 pairs of stereo headphones.

COMPLETE WITH AUDIO & POWER LEADS ....**AMA 1 – ONLY £14.99**

ADAPTOR FOR CONNECTING 2 PAIRS OF HEADPHONES......................**£2.99**

HIGH QUALITY STEREO HEADPHONES FOR MINIAMP 1 ETC. PER SET.**£4.99**

AMIGA MINI-AMP 2

MINI-AMP 2 is a neat stereo amplifier complete with two mini-speaker units which connects directly to your AMIGA. Perfect for use with mono tv's and monitors.

COMPLETE WITH AUDIO & POWER LEADS ....**AMA 2 – ONLY £19.99**

BARGAINS ON DISKS, BOXES, JOYSTICKS, DUSTCOVERS

FREE CATALOGUE WITH EVERY ORDER.

31/2" DISKS. QUALITY DS/DD BULK PACKED DISKS	**PER PACK of 10 ONLY £12.99**
31/2" DISKS: QUALITY DS/DD BULK PACKED DISKS	**PER PACK of 25 ONLY £28.49**
31/2" DISK BOX FOR 40 DISKS. SMOKED LID. LOCKABLE	**DSB 4 ONLY £7.99**
31/2" DISK BOX FOR 80 DISKS. SMOKED LID. LOCKABLE	**DSB 5 ONLY £9.99**
31/2" DISK BOX FOR 100 DISKS. SMOKED LID. LOCKABLE	**DSB 6 ONLY £11.99**
QUICKSHOT 2 JOYSTICK WITH AUTOFIRE	**QSJ 1 ONLY £7.99**
QUICKSHOT 2 TURBO, MICROSWITCHES & AUTOFIRE · GREAT	**QSJ 2 ONLY £13.99**
A500 DUSTCOVER, FLAMEPROOF MATERIAL	**ADC 1 ONLY £4.99**
A4 ANTISTATIC FOAM MOUSEPAD	**AFM 1 ONLY £4.79**

SHOWROOM NOW OPEN!

- **10 day money back guarantee if not 100% satisfied.**
- **All orders despatched promptly by recorded delivery.**
- **Express registered post and same day despatch — please add £1.95**

HOW TO ORDER

BY PHONE ▶ 0274 691115

OR FAX ▶ 0274 600150

WITH ACCESS OR VISA.

BY MAIL ▶ Send Cheque, Postal Order, PO Giro, or cash in registered letter. Eurocheque, Bankers Draft. Export sales our speciality, please add £1.00 additional postage and send payment in sterling only. Export Mastercard/Access and Visa Orders accepted by phone or FAX.

Trilogic, Dept. Y.A. Unit 1,
253 New Works Road, Bradford BD12 0QP.

AmigaBasic — Part 2

Allen Webb hacks a path deeper into the AmigaBASIC jungle

Before going further, a few comments concerning writing Basic code on wordprocessors. In keeping with most versions of Basic, AmigaBASIC has a SAVE command. If you follow the command with an A, the program will be saved as an ASCII file. For example:

```
SAVE "myfile",A
```

A file of this sort can be loaded into and used by a wordprocessor. This allows you to use the vastly better editing facilities offered by a WP to write and amend your programs. If you want to run the program, you simply use the MERGE command to load it. MERGE acts in the same way as the interpreter in that it converts all keywords to upper case so you can type the program in how you like. The 3D example given later in this article has been transfered from Basic into Scribble in this fashion.

This month I want to discuss the commands available for the control of program flow and to consider some of the elements of structured programming.

Let's get serious

It is worth the effort to develop a logical approach to programming. Several languages, such as Forth, Pascal and C all encourage structured programming. The usual approach is to first analyse the problem using a "top down" approach. This approach involves the splitting up of the task into sub-tasks and the splitting of sub-tasks into smaller tasks. Clearly there is a limit to this approach since the smaller the sub-tasks, the larger and more complex the program. You must seek a compromise position. It is frequently advised that subroutines and sub-programs should only be used for code called at several places in the program. I disagree. By adopting a modular approach of subroutines and sub-programs it is easy to debug each task and build up the program a piece at a time — rather like building a house from bricks!

Another important habit to develop is to make programs legible. Here are some hints:

- Use meaningful labels and variable names. A variable name such as ZD29 means little!
- Use plenty of explanatory remarks and comments.
- Get in the habit of using indented text to indicate different levels of looping or condition testing.

The important point is to be uninhibited. The Amiga offers a moderate amount of memory so you need not worry about using shorthand. Unlike the Basic on the earlier Commodore machines, there is no speed benefit to be accrued from the placing of Subroutines at the start of the program, so don't do it. Write the program in a clear and logical flow. I have seen many programs written in the so called "highly structured" BBC Basic which are unintelligible due to the use of shorthand and the highly obscure VDU statements.

Standard Basics normally offer FOR...NEXT loops, IF...THEN branching, GOTO, GOSUB, ON GOTO and On GOSUB. AmigaBASIC is somewhat more generous. The structures supported are IF...THEN....ELSE, GOTO, GOSUB...RETURN, ON...GOTO, ON...GOSUB, ON...ERROR GOTO, ON BREAK, ON COLLISION, ON MOUSE, ON MENU, ON TIMER, WHILE...WEND, FOR...NEXT, CALL. All of these tools are valuable aids to the writing of legible programs.

FOR...NEXT is is exactly as in normal Commodore Basic including the facility to omit the variable after the NEXT.

It is useful if you get in the habit of using indenting to indicate depth of looping. This aids the structure of your program and makes it more easily understood. Here is a trivial example of FOR...NEXT structures:

```
FOR A = 1 TO 10
   FOR B = 1 TO 10
      PRINT A,B,A*B
   NEXT B
NEXT A
```

The FOR instruction is left when the looping variable is greater than the specified end value.

GOTO structures are again the same in that they make an unconditional branch to a specified point in a program. AmigaBASIC doesn't use line numbers as other Basics but they can be used as labels. A more sensible approach is to use alphanumeric labels. Here is a trivial example:

```
Again:
PRINT "This is an infinite loop"
GOTO Again
```

GOTO is not a construct to use in structured programs. The heavy use of GOTO tends to be as a result of patches and kludges due to bad programming habits and can cause a program to be difficult to follow. It can be avoided if you plan the program structure properly.

An alternative method of looping is to use WHILE.....WEND. This construct loops continuously through the code bracketed by the WHILE....WEND so long has a given condition is true. You could, for example, test for a key press using:

```
WHILE INKEY$="" : WEND
```

The 3D demo uses WHILE to loop continuously until a key is pressed.

Alternatively, you can use a flag to decide if action is required. For example:

```
WHILE Flag=1
   PRINT "Flag is set"
WEND
```

The following example is equivalent to a FOR....NEXT construct:

```
WHILE X<10
   X=X+1
WEND
```

If only...

The central method of branching in Basic is the IF statement. In its simplest form, the instruction performs a test and if the test is true it performs an action. The action is either a jump or a sequence of instructions. For example:

```
IF FLAG=0 GOTO LOOP
IF FLAG>0 THEN FLAG=0
```

If you have a number of instructions in the sequence you can use a statement block. For example:

```
IF FLAG<0 THEN
   X=X+1
   Y=Y/3
   Z=X*Y
END IF
```

The END IF indicates the end of the statement block. Frequently you will wish to have alternative actions to test for. To allow this you may use ELSE:

```
IF Flag = 1 THEN
   PRINT “Flag=1”
   ELSE
      PRINT “FLAG<>1”
END IF
```

Probably the most important constructs are the subroutine and sub-program. Whilst they are similar in nature, they have subtle differences. A subroutine is generally a chunk of code which is called from a number of places in the program. It is normally intended to reduce unnecessary duplication of code. Consider the following code fragment which uses a hypothetical subroutine to print a message:

```
Message_string$ = “Hello” : GOSUB
Print_mess
Message_string$ = “There” :
GOSUB Print_mess
Message_string$ = “My name” :
GOSUB Print_mess
Message_string$ = “is Fred” :
GOSUB Print_mess
END

Print_mess:
   Print Message_string$
   BEEP
   RETURN
```

The operation of GOSUB is as in other Basics. The address of the GOSUB is placed on the stack and the program execution transfers to the start of the subroutine. Once RETURN is reached the GOSUB address is pulled from the stack and the execution returns to the instruction after the GOSUB.

A rather more complex beast is the sub-program. This is a separate chunk of code which has its own identity and, if you want, its own variables. I won't discuss sub-programs in great depth since the manual devotes a fair number of pages to them. I will, instead, give an outline of their use. The syntax of a sub-program is as follows:

```
SUB subprogramname (parameter list)
STATIC
      ...
      ...
      ...
END SUB
```

A sub-program is entered by CALL or simply by reference to its name. The syntax for CALL is:

CALL subprogramname (parameter list)

The parameter list indicates which parameters are to be passed. STATIC, for some reason, is mandatory. Try this example:

```
Start:
   CALL Pig
   Var=10
   PRINT Var
GOTO Start

SUB Pig STATIC
   PRINT Var;
   Var=Var+5
END SUB
```

This example shows that the two variables Var while having the same name have independent existences. The value printed from the sub-program progressively increases by 5 while the other value remains at 10. If you want to alter variables both in the sub-program and the main program, you must declare the variables with a SHARED declaration in the sub-program. I prefer the descriptor GLOBAL for shared variables and LOCAL for static variables rather than the AmigaBASIC syntax, but you can't have everything.

When you transfer a variable to a sub-program, its value is adopted by the corresponding variable in the SUB declaration. For example:

```
CALL Print_String (Mess_string$)
END
SUB Print_String (P$) STATIC
   Print P$
END SUB
```

The value is moved into P$ for the purposes of the sub-program. It is also possible to protect variables from being altered in the sub-program by bracketing the variable in the CALL parameter list. There are a number of other options but I'll leave you to read the manual for those. So what is the value of the sub-program? Well the obvious benefits are twofold:

- They allow you to build the program from a series of building blocks which may be individually tested and de-bugged.
- You can build a library of self-contained sub-programs which can be used, as required, in future programs.

The remaining important control structures are ON....GOTO and ON....GOSUB. The syntax of these commands is:

ON expression GOSUB line list
ON expression GOTO line list

The expression is evaluated and the integer part taken as a pointer to determine the destination of the branch. If, for example, the expression yields a value of 2, then the destination is the second label on the list. Here is an example:

```
ON Val GOSUB one,two,three
END
one:
   PRINT “Val equals one”
RETURN
two:
   PRINT “Val equals two”
RETURN
three:
   PRINT “Val equals three”
RETURN
```

The GOTO or GOSUB action works as before.

Trapping events

In addition to these structures, there are a number of “event trapping” instructions. These act on a stimulus such as a timer call, a mouse action or an error. Some are linked to graphics and WIMP actions, which I will deal with another time.

ON BREAK. This command detects CTRL-C presses or stop calls from the menu and jumps as instructed. This allows you to prevent the user stopping the program. The instruction is supported by three other commands:

BREAK ON This enables the event trapping.

BREAK OFF This disables the event trapping

BREAK STOP This suspends event trapping so that the GOSUB portion is ignored.

Here is a trivial example:

```
BREAK ON
   loop:
      ON BREAK GOSUB loop2
      a$=inkey$
      IF a$=“” THEN loop
   END

   loop2:
      BEEP
   RETURN
```

This disables stop but allows you to stop the program with any key press.

ON ERROR GOTO. This detects an error and jumps to the specified line. It allows you to establish the nature of the error, print a helpful message and return to the place where the error occured via a RESUME command.

ON TIMER. Here the timer is used to execute a routine at regular intervals.

The syntax is:

ON TIMER(n) GOSUB label

As with BREAK, the detection system is controlled by TIMER ON, TIMER OFF and TIMER STOP commands.

The 3D example is intended to show some of the concepts I have described.That's about it this time, next issue I will enter the realm of graphics. *YA*

"THE MAIN EVENT"

DEPT (YC) 61 STAFFORD STREET, HANLEY, STOKE-ON-TRENT,
STAFFORDSHIRE ST1 1LW

24 HOUR ANSWER SERVICE. TEL: 0782 281544

AMIGA	
FIRE AND FORGET	17.50
HOT SHOT	£13.90
PLATOON	£17.50
GRYSOR	£17.50
COMBAT SCHOOL	£17.50
GREMLINS HOTS HOTS	13.90
CHUBBY GRISTLE	13.90
GREAT GIANA SISTERS	17.99
BETTER DEAD THAN ALIVE	13.90
BEYOND ICE PALACE	17.50
BIONIC COMMANDO	17.50
BEYOND ZORK	17.50
BUREACRACY	24.99
EBO??STAR	17.50
FIREPOWER	17.50
G L FOOTBALL	17.50
FLINSTONES	13.90
TERRAMEX	13.90
WORLD GAMES	17.50
BAD CAT	17.50
MOUSE TRAP	10.50
THREE STOOGES	21.00
JOE BLADE	6.99
JET	28.50
GARRISON 2	17.50
GARRISON	17.50
GOLDRUNNER	17.50
GRAND SLAM TENNIS	21.00
HUNT RED OCTOBER	17.50
INDOOR SPORTS	17.50
MEAN 18 GOLF	21.00
SIDEWINDER	6.99
STARGLIDER	17.50
STRIP POKER 2	10.50
WESTERN GAMES	13.99
WINTER GAMES	17.50
WINTER OLYMPIAD 88	13.90
WIZBALL	17.50
XENON	13.90
XR 35	6.99
SPIN WORLD	13.90
SPACE STATION	6.99
SPACE QUEST 2	13.90
SKY BLASTER	13.90
ROMANTIC ENCOUNTER	17.50
ROLLING THUNDER	17.50
PROTECTOR	6.99
POWERPLAY	13.90
POWER STYX	13.90
POLICE QUEST	13.90
PINK PANTHER	13.90
PINBALL WIZARD	13.90
P. BEARDSLEY FOOTBALL	13.90
MIND FIGHTER	17.50
LITTLE DRAGON	13.90
LEATHERNECK	13.90
LEATHER GODDESS	21.00
JINKS	17.50
GUNSHOOT	13.90
FLIGHT SIM 2	28.99
FLINSTONES	13.90
FOOTBALL MANAGER 2	12.99
ROAD BLASTERS	17.50
ALIEN SYNDROME	13.90
CORRUPTION	17.50
FIRE STATION	6.99
FINAL MISSION	13.90
EXPLORA	28.00
EMERALD MINE	13.90
EBONSTAR	17.50
DROID 2	13.90
CRYSTAL HAMMER	13.90
BUBBLE BOBBLE	13.90
BUBBLE GHOST	13.90
BAD CAT	17.50
PHANTASM	13.99
CARRIER COMMAND	17.50
ZOOM	13.90
WORLD DARTS	6.99
FIRE 'N' FORGET	13.90
IKARI WARRIORS	17.50
ALIEN	13.90
SREET FIGHTER	17.50
CITY DEFENCE	10.50
SUPER HUEY	13.90
THUNDER BOY	10.50
AAARGH	13.90
BATTLESHIPS	13.90

AMIGA	
SPACE STATION	6.95
WAR ZONE	6.95
FIRE BLASTER	6.95
PROTECTOR	6.95
ENFORCER	6.95
PLAYHOUSE STRIP POKER	9.99
ONE ON ONE	14.90
ARCHON ??	14.90
7 CITIES OF GOLD	14.90
STATION FALL	13.99
MOON MIST	13.99
SUSPECT	13.99
BALLYHOO	13.99
MIND FOREVER VOYAGING	13.99
DEADLINES	13.99
LITTLE COMP PEOPLE	13.99
PORTALL	13.99
MINDHSADOW	13.99
CUT THROATS	13.99
ZORK 2	13.99
ZORK 3	13.99
ENCHANTER	13.99
SORCERER	13.99
INFIDEL	13.99
STAR CROSS	13.99
HACKER	13.99
PUB POOL	6.99
AMIGES	10.50
ANNALS OF ROME	13.90
ARENA	21.00
ART OF CHESS	17.45
BALANCE OF POWER	21.00
BACKLASH	13.90
BARBARIAN (PALACE)	13.90
BARBARIAN (PSY)	17.50
BLACK JACK ACADEMY	17.50
BLASTER BALL	6.99
BMX SIMULATOR	10.50
CASINO ROULETTE	13.90
CHALLENGER	6.99
CRUNCH FACTORY	6.99
DEMOLITION	6.99
FEUD	6.99
MUSIC STUDIO	28.99
ANALYZE	35.00
BBS PC	89.00
ORGANIZE	89.00
PUBLISHER 1000	179.00
SCRIBBLE	89.00
T.V. TEXT	89.00
ZUMA FONTS 1	29.00
ZUMA FONTS 2	29.00
ZUMA FONTS 3	29.00
GRAPHIC TABLET	34.99
MI-AMIGA FILE	89.00
LATTICE C	119.00
LATTICE C PROFIT	279.00
LISP	139.00
PASCAL	79.00
PRO SOUND DESIGNER	69.99
MUSIC STUDIO	29.99
FOOTBALL DIRECTOR 2	11.00
FOOTBALL MANAGER 2	10.50
FERNAND?? MUST DIE	17.45
ROCKET RANGER	20.00
POOL OF RADIANCE	17.45
MENICE	13.90
NETHERWORLD	13.90
CARRIER COMMAND	17.45
FEDERATION OF FREE TRADERS	19.49
PAC MANIA	13.90
WHERE TIME STOOD STILL	11.95
BIONIC COMMANDOS	17.45
STAR GLIDER	17.45

COMMODORE 64	
JACK THE NIPPER	2.50
LAST MISSION	2.50
DECEPTOR	2.50
FIELDS OF FIRE	2.50
KNIGHT GAMES II	2.50
OLTAPOLIS	2.50
QUARTET	2.50
SUPER SPRINT	2.50
FIRE TRAP	2.50
BACK TO FUTURE	2.50
SPORTING CLASSICS	2.50
ACE	2.50
MYSTERY OF NILE	2.50
ON THE TILES	2.50
SIDEWIZE	2.50
SCARY MONSTERS	2.50
SOLOMON'S KEY	2.50
RYGAR	2.50
TRANTOR	2.50
BLADE RUNNER	2.50
CAPTAIN AMERICA	2.50
FIVE STAR GAMES 2	2.50
STAR PAWS	2.50
MARBLE MADNESS	2.50
ARFIC	2.50
LEADER BOARD	2.50
SCOOBY DOO	2.50
PING PONG	2.50
TAG TEAM WRESTLING	2.50
GAUNTLET	2.50
GAUNTLET DEEPER DUNG	2.50
DONKEY KONG	2.50
BREAKTHRU	2.50
SHORT CIRCUIT	2.50
BOMBJACK II	2.50
FOOTBALL MANAGER	3.95
SUPER SUNDAY	3.50
HEAD COACH	3.50
PRESIDENT	3.50
JET SET WILLIE 2	3.50
SOLDIER ONE	3.50
SHAOLINES ROAD	3.50
URIDUM/PARADROID	3.50
PSI TRADING	3.50
RAMPARTS	2.50
PARALAX	3.50
LEGEND OF KAGE	3.50
HACKER 2	3.50
V	3.50
HEAD OVER HEALS	3.50
ZAXXON	3.50
MULE	3.50
SPACE SHUTTLE	3.50
PITFALL	3.50
KENSINGTON	3.50
ROBOTRON	3.50
ZORRO	3.50
WIZARD	3.50
SHOCKWAY RAIDER	3.50
CRYSTAL CASTLE	3.50
JEWELS DARKNESS	3.50
BOULDERDASH CON KIT	3.50
ROAD RUNNER	3.50
RENEGADE	3.50
STAR GAMES II	3.50
FLIGHTPATH 737	3.50
THUNDERCATS	3.50
BUBBLE BOBBLE	3.50
BUGGY BOY	4.75
SPACE MARRIOR	4.75
ADICTIBALL	4.95
FRANKENSTAIN	2.50
I ALIEN	2.50
VENGEANCE	2.50
DISCOVERY	2.50
JET BOYS	2.50
TIME FIGHTER	2.50
THUNDER CROSS	2.50
MANDROID	2.50
OINK	2.50
WOLF MAN	2.50
STRATTON	2.50
TRAXXION	2.50
BOOK OF DEAD	2.50
LIFE FORCE	2.50
BIG TROUBLE IN CHINA	2.50
SUPER STAR PING PONG	2.50
PARADROID/URIDIUM	2.50
THE YOUNG ONES	2.50

COMMODORE 64	
PINK PANTHER	6.99
SKATE CRAZY	6.99
HERCULES GREMLIN	6.99
BLOOD BROTHERS	6.99
GREMLINS HOTS HOTS	6.99
G.I. HERO	6.99
LAST NINJA II	8.99
BIONIC COMMANDO	6.99
EVERY SECOND COUNTS	5.59
GREAT GIANA SISTERS	6.99
ROAD BLASTERS	6.99
DREAM WARRIORS	6.99
MARRAUDER	6.99
MATCHDAY II	6.25
WIZARD WARS	6.99
KIDSPLAY COMPILATION	6.99
SALAMANDER	6.25
MICKEY MOUSE	6.99
OBSOLATOR	6.99
ALIEN SYNDROME	6.99
HOPPING MAD	6.99
THE VINDICATOR	6.25
EDDIE EDWARDS SUPERSKI	6.99
PLATOON	6.99
RASTAN	6.25
THUNDERCATS	3.95
COMBAT SCHOOL	6.25
GAYSTOR	6.29
D.T. OLYMPIC CHALLENGE	6.95
TARGET RENEGADE	6.25
KONAMI ARCADE COLLECTION	6.99
STREETFIGHTER	6.99
SPECIAL FX	6.99
ARKANOID	6.25
IMPOSSIBLE MISSION II	6.98
PREDATOR	6.99
BEYOND ICE PALACE	.6.99
MAGNETRON	6.25
GAUNTLET II	6.98
ROLLING THUNDER	6.99
SUPER HANG ON	6.99
BLACKLAMP	6.25
CYBERNOID	6.99
DARKSIDE	6.95
EMPIRE STRIKES BACK	6.99
VENOM STRIKES BACK	6.99
IKARI WARRIORS	6.95
FIREFLY	6.25
OUTRUN	6.99
720	6.99
CHERNOBYL SYNDROME	6.90
DAN DARE II	6.25
CARD SHARKS	6.95
MINI PUTT	6.95
NORTH STAR	6.99
OOPS!	6.99
COMPUTER HITS 5	2.50
ELITE COLLECTION	9.95
GOLDEN OLDIES	2.50
KNIGHT RIDER	6.25
DRAGONS LAIR	6.25
10 GAMES BY GREMLIN	6.95
FERNANDEZ MUST DIE	6.90
NATO ASSAULT COD	9.00
TRAZ	6.95
CRAZY CARS	6.95
YETI	6.95
CROSSWIZ	6.25
CHARLIE CHAPLIN	6.99
NOW GAMES 3	2.50
VIXEN	6.95
TAITO COIN OPS	8.90
THE FLINSTONES	6.99
THE ULTIMATE PACK	5.90
ACTION PACK II	2.99
GUTZ	6.2?
CHUBBY GAISTLE	6.98
INTENSITY	6.90
ECHELON	8.99
BUTCHER HILL	6.50
SHACKLED	6.99
BUGGY BOY	3.9?
MASK	2.5?
TYPHOON	6.99
BARDS TALE	6.95
TRACKSUIT MANAGER	6.99
HERD OF THE LANCE	6.95
MERCENARY	6.95
TECHNO COP	6.95

Please cross PO/Cheques and make payable to: THE MAIN EVENT.
To help maintain our low prices we would be obliged if you could include a first class stamp with your order, although this is NOT compulsory. Free Post & Packaging. Overseas please add 75p per tape.
We apologise for any alteration or omissions since going to press. The Main Event is a division of Software City.

Star Ray

Karen Young asks, has Logotron hit the big time with this version of Defender — or maybe not...?

Logotron is a strange company, catering more or less exclusively for the BBC Market until last year, they have decided to go 16-bit with the launch of this game and another called *Star Goose*. This review is about the former, and it did bring back a few memories of my misspent youth, hacking away in arcades, trying to complete yet another wave of this stupid game.

For *Star Ray* is yet another Defender clone. The gameplay is almost identical, but the graphics... The programmers have taken a great deal of care and attention over the look, and I think once you see this game in action on an Amiga near you, you'll be impressed.

With a very nice perspective system that makes the backgrounds move at different speeds in relation to their distance, *Star Ray* — your ship — speeds into action, ready to zap the bloody hell out of the aliens as soon as they materialise above the surface of your terrain.

The monsters are fairy standard fare; On level one they look like cast-offs from earlier Psygnosis games. Level two gives us the temperate climate of a jungle planet, and so on...

What more can I say? The ships carry letters that, once shot, give you additional bonuses (such as added shields, more destructive firepower, better manoeuvrability and additional "smart bombs") but all in all *Star Ray* is just another Defender with a few more bells and whistles.

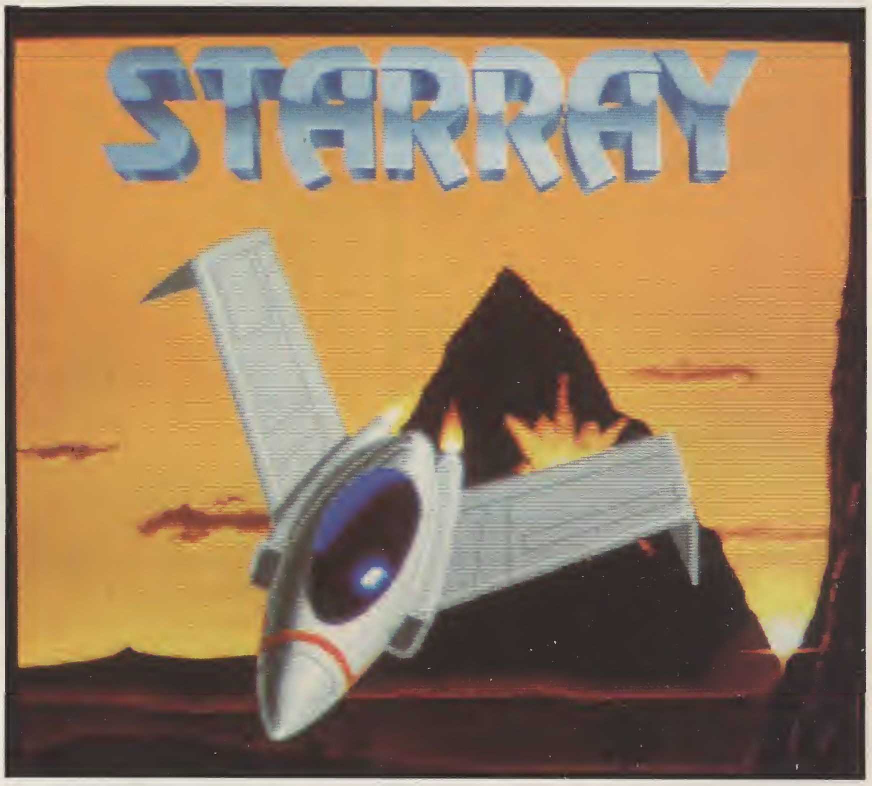
Not that I mind — the game is saved from the "format and use elsewhere" pile by virtue of the beautiful graphics, there's plenty of shading, some very nice perspective work (as I mentioned earlier) and the controls are logical and smooth — use a joystick and you are in for a headache but use a mouse, and things are vastly improved. The game took a surprising turn for the mega-terrific when Stuart Cooke, the editor of this illustrious tome, brought in his trackball.

With my hands on his ball I suddenly understood the meaning of *Power*. If *Marble Madness* wasn't a good enough reason to go out and buy a trackball, then this must really be the clincher. His ball is quite a small model (from one of his American trips — I think), but the smaller balls have their advantages, being more controllable, and easier to stop when they look as if they are going to overshoot. Big trackballs have a momentum problem that require a lot of pushing to get them down to speed.

The sound effects are probably the worst part of the game. My best description must be "from the sublime to the bland". Sure, the lasers sound like what you would expect from lasers, and the explosions are, well, pretty explosive, and the little jingle is annoying enough to make you reach for the volume button, but they really do not push the Amiga to its limits — a shame. A lot of games let themselves down in this area — *Star Ray's* only sin is of being merely competent, not brilliant, as all games should be.

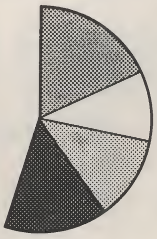
I have often wondered how many Amiga players were willing to use a trackball with their games (instead of a mouse). From the games playing side, it would seem more logical to get one between your fingers — they take up less desk space and they offer an alternative to bashing out on the mouse. Some of the office mice are well and truly knackered after playing games all day (makes you wonder how magazines are put together doesn't it?), but Stuart's ball seems remarkably easy to play with — I think I will just have to go out and get my own! A good start for Logotron though, their game for the BBC B was excellent, but with the unlimited scope of the Amiga the company have merely brushed the surface, which is a damn-sight deeper than other companies,merely happy with simply porting across software from the crappy old Atari ST.

Star Ray is an excellent game. I especially liked the level save system whereby you can start at any level you like providing you have played it before — this solves the problem of passwords and having to remember them. My copy stopped me from playing if I left the game running for ten minutes or so. I hope this isn't going to be the case with the production version as you frequently have to leave the Amiga running while you answer the telephone or make a cup of tea! A nice start though, and I look forward to seeing what else Logotron has planned. Ok? *YA*



STAR RAY

Title: **Star Ray**
Supplier: **Logotron**
Dale's Brewery
Guydir Street
Cambridge
CB1 2LJ
Tel: **0223 323696**



▩ Graphics: **18**
☐ Sonics: **10**
▨ Gameplay: **12**
■ Value: **15**

LAN COMPUTER SYSTEMS

01-597 8851

£ 229.00 INC VAT
INCLUDING ALL CONNECTING LEADS

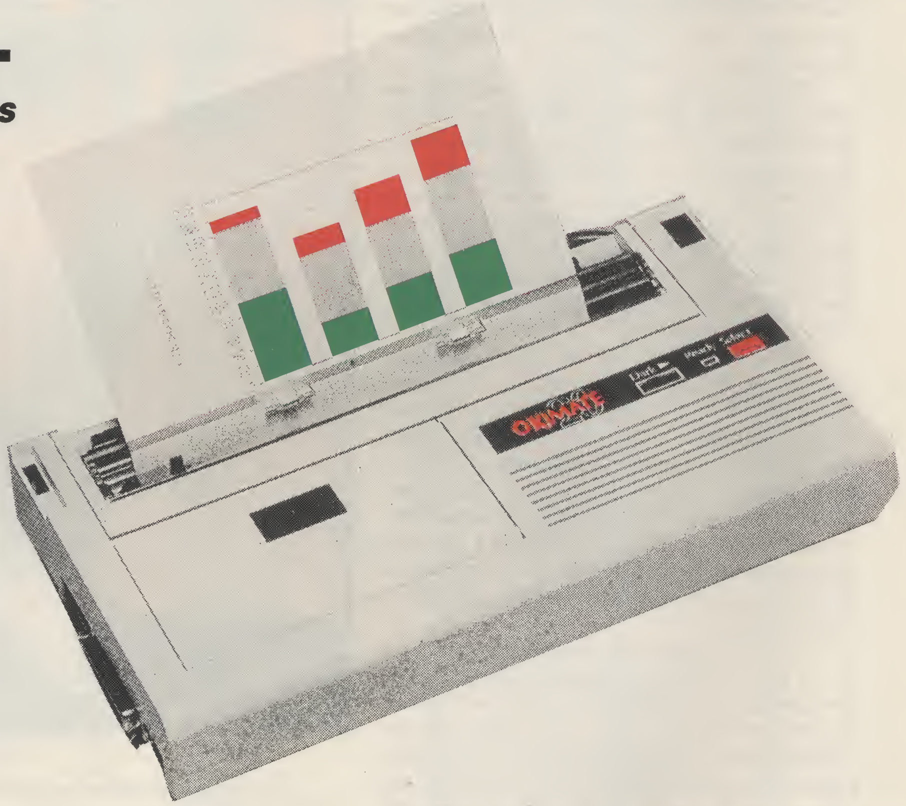
SPECIAL SOFTWARE OFFERS

	R.R.P.	Our Price
Ports of Call	39.95	15.00
Three Stooges	29.95	16.00
Aaargh	19.95	11.00
Palace Barbarian	19.95	11.00
Leatherneck	19.95	11.00
Pinball Wizard	19.95	11.00

Prices inclusive of VAT, please add £1.00 for delivery UK + Europe.

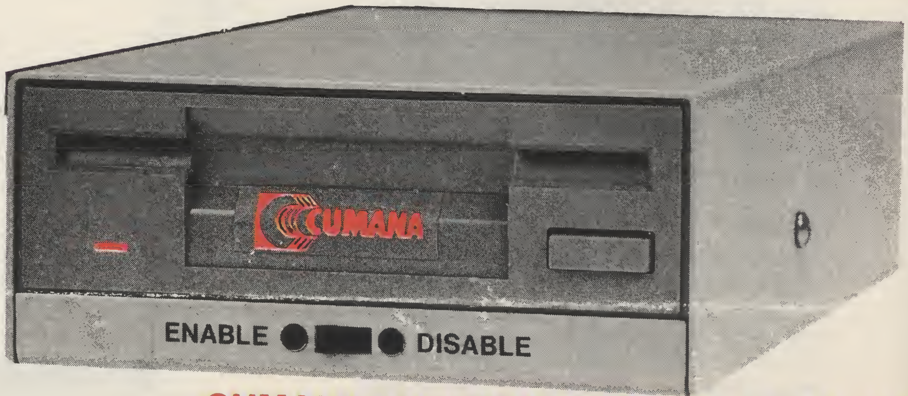
Transformer £19.95
5¼ Disks box of ten £5.75

OKIMATE 20
ONLY
£139.00 INC VAT
INCLUDES ALL CONNECTING LEADS



★ LC 24/10
£279 + VAT

WHY NOT TREAT YOURSELF
TO TEN D/S 3½ DISKS
FOR AN EXTRA £10.00 INC VAT.



AMIGA CUMANA

CUMANA 3½" DISK DRIVE £99.95 INC VAT

5¼ INCH DISK DRIVE EITHER AMIGA 2ND DRIVE
OR 2ND DRIVE FOR PCI £109.95 INC VAT

LAN COMPUTER SYSTEMS

SHOW ROOM OPEN MON TO SAT 10.30am to 5.30pm

ONE OF EUROPES LARGEST AMIGA CENTRES WE ONLY SELL GENUINE U.K. SPEC AMIGA

LAN LAN

LAN AMIGA SOFTWARE PACK

AMIGA 500

Free 20 programmes condensed onto 2 disks for your convenience.
FREE MOUSE MAT *FREE*DUST COVER *FREE* JOYSTICK
£100 OF SOFTWARE WITH YOUR AMIGA

YES YOU CHOOSE THE SOFTWARE YOU WANT WITH YOUR AMIGA (£100 RRP) IF YOU WISH TO ORDER A TV MODULATOR THEN YOU WILL HAVE **£75.00** TO SPEND ON SOFTWARE

£419 INC. VAT

AMIGA COLOUR PRINTER PACKAGE

AMIGA 500. Free Mouse Mat, Free Dust Cover, Free Deluxe Paint 1, Free 20 programmes condensed onto 2 disks for your convenience.

Okimake Colour Printer
The Work (Word Processor, database + worksheet)
SAVE £100

£478 + VAT

★ MONITORS ★

1084 High Res Col monitor	£199.00 +VAT
if purchased with computer	£189.00 +VAT
Philips 8833 monitor	£219.00 +VAT
if purchased with computer	£209.00 +VAT
High resolution monochrome monitors from	£69.00 +VAT
Nec Multisinc II	£469.00 +VAT

★ ACCESSORIES ★

A501 RAM expansion	£129.00 inc VAT
Mouse mat	£4.95 inc VAT
Amiga dust cover	£4.95 inc VAT
Philips/1084 dust cover	£6.95 inc VAT
3½" disk drive cleaning kit	£7.95 inc VAT
TV modulator	£24.15 inc VAT
10 D/S 3.5 disks	£14.95 inc VAT
20 D/S 3.5 disks in 80 capacity lockable box	£29.95 inc VAT

★ DRIVES ★

Cumana CAX354 with on/off switch	£99.95 inc VAT
Supra 20mb hard disk	£400.00 inc VAT

LANSOFT CLUB

The Club for St and Amiga users everywhere all prices include VAT

★ Special Introductory Prices ★
★ Membership — £20 — (per year) ★
★ Save up to 40% off Games Software ★
★ Up to 30% — Off Application Software ★
Special Club Prices on Cumana Disk Drive £89.00 inc VAT
★ A501 Ram Expansion £99.99 inc VAT ★

01 597 8851
1063 HIGH RD,
CHADWELL HEATH,
ROMFORD, ESSEX, RM6 4AU

★ PRINTERS ★

Citizen 120D	£99.00 +VAT
Panasonic KXP 1081	£145.00 +VAT
Star LC 10 Printer	£199.00 +VAT
Star LC 10 Printer, Stand, Paper & Cable	£209.00 +VAT
Nec 24 Pin Printer	£299.00 +VAT
Epson LX 800	£169.00 +VAT
256K Printer Buffer	£160.00 +VAT
Printer Cable	£13.00 +VAT
Box of Paper 2000 Sheets	£13.00 +VAT
Printer Stand	£9.95 +VAT
Printer Stand	£26.00 +VAT
Printer Ribbons from	£3.00 +VAT
New Star LC24-10	£299.00 +VAT

★ SPECIAL OFFER ★

In stock
STAR LC 10 COLOUR
£233.00 + VAT

OKIMATE 20 COLOUR PRINTER
includes cable, paper roll holder, colour paper holder & a host of other accessories
£139.00 + VAT

OKIMATE 20 COLOUR PRINTER
£139.00 inc VAT

All prices exclusive of VAT & delivery unless otherwise stated

HOW TO ORDER

Enclose letter with cheque, postal order or credit card number for amount including VAT and delivery charge. Credit card holders may order by telephone. Contact sales desk. Dispatch normally in 24 hours.

Prices current at time of going to press. Subject to availability.

Official Orders from Educational Establishments, Local Authorities and Government departments welcome. Contact Education Department.

FAX: 01-590 6057. Telex: 995548.
Telex: 995548.

★ AMIGA B2000 ★

Internal Genlock	Phone
8mb RAM Board Powlated to mb	Phone
B2000	£849.00
2mb RAM BOARD	£399.00
XT Bridge Board	£399.00
20mb hard disk	£399.00
A501 RAM expansion now in stock	

APPLICATIONS SOFTWARE

The Works All For £69.00 inc. VAT

A Wordprocessor + Database and Spreadsheet

The Works is A Three in One Pack

Scribble 2 is the Word Processor
Analyse 2 is the Spreadsheet
Organise is the Database

TV TUNER

Turn your monitor into a TV. 12 Programmes selection complete with all cables + aerial switchable between TV + computer.
ONLY £59.00 + VAT
if ordered with monitor
£55.00 + VAT

AMIGA.LIB RULES OK!

Margaret Stanger guides you through the methods of using the Amiga library files

The Amiga provides a wealth of software support that enhances the capabilities of its powerful hardware. This software support takes the form of *Library Routines*, which are special Amiga resource files which are bound to your program dynamically at run time.

The routines range over several levels from primitive device management or graphics drawing routines to Intuition, which handles overlapping window displays and several input and output devices.

There are several libraries available for use in your own applications, each containing several routines. Associated with each library is a special **.FD** file that describes the parameters that the routines require, and the registers used in assembly language.

The name of each routine is used by the compile and link programs to work out its offset from the library base. The exec library base is always in the same place, location 4, or the external reference _AbsExecBase can be used.

Accessing A Library

To access a library routine you must first open the library that contains that routine. This will give you the library base. As many as five libraries can be open at any one time. Once the library is open, its routines can be called as if they were subprograms or your own machine language routines.

In C you can call the routine by name, using the normal C language conventions. For assembly language the library base must be in register a6, if it is in any other register the routine could overwrite it. The name of the routine is declared as an external reference and used as if it were a straightforward offset from the library base, the linking program taking care of the rest.

In this series I hope to show how I have used the Amiga library routines to access the Amigas graphics, sound and the ports. The source code for each program can be entered in either C or Assembler, both will be given. I have not dwelt on the syntax of the source code or how to compile it into a final program as these subjects are already covered by the *Your Amiga* series on C and on Assembler and depend to a large extent upon the assembler/compiler that you are using.

There are several libraries that are internal (for example intuition,dos,exec, graphics) they may differ slightly with the operating system. Other libraries, (for example diskfont) are diskbased and are expected to be in the libs directory of the system disc. A complete list of the library routines can be found in the fd.files directory, which is usually on the workdisk. The 1.2 version fd.files on my Workbench disc differ slightly from the 1.1 version that came with the C and Assembler, so there are a few library routines somewhere

DOS LIBRARY SUMMARY

Name	Inputs	Registers	Result in D0	Effect
Open	(name,accessMode)	(D1/D2)	pointer to filehandle	Opens a file for input or output -30 ; 1005=old 1006=new
Close	(file)	(D1)	-	Closes a file for input or output -36
Read	(file,buffer,length)	(D1/D2/D3)	length read	Reads bytes of data from a file -42
Write	(file,buffer,length)	(D1/D2/D3)	length written	Writes bytes of data to a file -48
Input	()		filehandle	Identifies programs initial input file handle
Output	()		filehandle	Determines programs initial output filehandle
Seek	(file,position,offset)	(D1/D2/D3)	old position	Moves to logical position in a file
DeleteFile	(name)	(D1)	boolean success or fail	Deletes file
Rename	(oldName,newName)	(D1/D2)	boolean	Renames a file
Lock	(name,type)	(D1/D2)	pointer to lock	Locks a file
UnLock	(lock)	(D1)	-	Unlocks a file
DupLock	(lock)	(D1)	pointer to new lock	Duplicates a lock
Examine	(lock,fileInfoBlock)	(D1/D2)	boolean	Examines directory or file
ExNext	(lock,fileInfoBlock)	(D1/D2)	boolean	Examines the next file in the directory
Info	(lock,parameterBlock)	(D1/D2)	boolean	Returns information about a disk
CreateDir	(name)	(D1)	pointer to lock	Creates a new directory
CurrentDir	(lock)	(D1)	pointer to old lock	Makes this diectory the current one
IoErr()			boolean	Indicates if an error has occurred in some routines
CreateProc	(name,pri,segList,stackSize)	(D1/D2/D3/D4)	identifier	Creates a process
Exit	(returnCode)	(D1)	-	Exits from program
LoadSeg	(fileName)	(D1)	pointer to segment	Loads a module into memory
UnLoadSeg	(segment)	(D1)	-	Unloads a segment loaded by LoadSeg
DeviceProc	(name)	(D1)	pointer to process	Returns an identifier
SetComment	(name,comment)	(D1/D2)	boolean	Sets a comment on a file or diectory
SetProtection	(name,mask)	(D1/D2)	boolean	Sets read/write/delete/execute file protection
DateStamp	(date)	(D1)	-(date put in input array)	Obtains the internal time
Delay	(timeout)	(D1)	-	Delays a specified length of time
WaitForChar	(file,timeout)	(D1/D2)	boolean	Success if character arrives in specified length of time
ParentDir	(lock)	(D1)	pointer to lock	Obtains parent of directory or file
IsInteractive	(file)	(D1)	boolean	Determines whether the file is connected to a virtual terminal
Execute	(string,file,file)	(D1/D2/D3)	boolean	Executes a CLI command

that I can only access from Basic at the moment!

The Project

The program presented here uses a few of the routines from the dos and exec libraries, but there are many more available. I have included a list of all the 1.1 library dos and exec routines that were documented. For each one I have listed the input, input registers, output (where applicable) and a short description of the effect of the routine.

The program is designed to give an introduction to some of the routines in the dos and exec libraries, and the source code is provided in C and Assembler. The C source code file (named "part1.c") was compiled and linked using Lattice C version 3.03. The assembly source code file (named "part1.asm") was assembled and linked using the Metacomco assembler version 10.178.

In Assembler,when a library is opened, the pointer is stored by the program. If a library routine is declared as an external reference (XREF),the linker can calculate its offset from this pointer. When writing in C, the compiler handles all these details, so there is no need to state which routines you are going to use, or to which library they belong.

When run, the program opens the dos library, graphics library, intuition library and diskfont library, and waits for a few seconds. If the program ends straight away instead of waiting, then one or more of the libraries failed to open. This is more likely to be the diskfont library as it is diskbased rather than internal. It should be present in the libs directory of the boot disc, otherwise this library will not open.

The program looks for the source code text file to read in, if the file is not found in the current directory of the program will exit at this stage. Any text file could be used, provided the filename is included correctly in the source code. If all is well the first 1024 characters of the file will be displayed on the screen before the program frees the memory, closes the libraries and exits.

More To Come

The next part of this series will feature one or two of the many routines from the intuition library, and will include a library summary.

```
          EXEC LIBRARY SUMMARY

Name             Inputs           Registers    Results in d0  Effect

Libraries:

AddLibrary      (library)          (A1)        -              Adds a library to the system
RemLibrary      (library)          (A1)        -              Removes a library
OpenLibrary     (libName,version)  (A1,D0)     librarypointer Opens a library                          -552
CloseLibrary    (library)          (A1)        -              Closes a  library                        -414
SetFunction(library,funcOffset,funcEntry)(A1,A0,D0)  old function  Changes  function vector in a library   -420
SumLibrary      (library)          (A1)        -              Computes  and check library checksum     -426

Memory allocation:

Allocate        (freeList,byteSize)     (A0,D0)     memory block   Allocates a block of memory using the FreeList
Deallocate(freeList,memoryBlock,byteSize)(A0/A1,D0) -              Deallocates  a memory block
AllocMem        (byteSize,requirements) (D0/D1)     memory block   Allocates a memory block                      -198
AllocAbs        (byteSize,location)     (D0/A1)     -              Allocates a memory block from an absolute address -204
FreeMem         (memoryBlock,byteSize)  (A1,D0)     -              Frees a memory block allocated with AllocMem  -210
AvailMem        (requirements)          (D1)        size           Finds amount of memory available              -216
AllocEntry      (entry)                 (A0)        list           Allocates many regions of memory
FreeEntry       (entry)                 (A0)        -              Frees many memory regions
Special functions

InitStruct      (initTable,memory,size) (A1/A2,D0) -              Initialises memory from a table
MakeLibrary(funcInit,structInit,libInit,dataSize,codeSize)(A0/A1/A2,D0/D1)

Interrupts

Disable         ()                                  -              Disables interrupts                       -120
Enable          ()                                  -              Enables interrupts following a disable    -126
Forbid          ()                                  -              Prevents task rescheduling and multitasking
Permit          ()                                  -              Permits task rescheduling and multitasking following a forbid
SetSR           (newSR,mask)            (D0/D1)     old SR         Gets or sets processor status register
GetCC           ()                                  conditions     Gets condition codes
SuperState      ()                                  old system stack Enters supervisor state
UserState       (sysStack)              (D0)        -              Enters user state
SetIntVector    (intNumber,interrupt)   (D0/A1)     old vector     Sets a system interrupt vector
AddIntServer    (intNumber,interrupt)   (D0/A1)     -              Adds an interrupt server to the system
RemIntServer    (intNumber,interrupt)   (D0/A1)     -              Removes an interrupt server
Cause           (interrupt)             (A1)        -              Causes a software interrupt               -180

Lists:

Insert          (list,node,pred)        (A0/A1/A2)  -              Inserts node to list
AddHead         (list,node)             (A0/A1)     -              Inserts node at head of list
AddTail         (list,node)             (A0/A1)     -              Appends node to tail of list
Remove          (node)                  (A1)        -              Removes node from list
RemHead         (list)                  (A0)        -              Removes head node from list
RemTail         (list)                  (A0)        -              Removes tail node from list
```

▶▶

Enqueue	(list,node)	(A0/A1)	-	Appends node to system queue	
FindName	(list,name)	(A0/A1)	node	Finds a system list with a given name	
Tasks:					
AddTask	(task,initPC,finalPC)	(A1/A2/A3)	-	Adds task to the system	
RemTask	(task)	(A1)	-	Removes task from the system	
FindTask	(name)	(A1)	task	Finds a task with a given name or the current task	
SetTaskPri	(task,priority)	(A1,D0)	old priority	Gets and set the priority of a task	
SetSignal	(newSignals,signalSet)	(D0/D1)	old signal	Defines the state of this tasks signals	
SetExcept	(newSignals,signalSet)	(D0/D1)	old signal	Defines signals to cause exceptions	
Wait	(signalSet)	(D0)	-	Waits for one or more signals	
Signal	(task,signalSet)	(A1,D0)	-	Signals a task	
AllocSignal	(signalNum)	(D0)	signal number	Allocates a signal bit	
FreeSignal	(signalNum)	(D0)	-	Frees an allocated signal bit	
AllocTrap	(trapNum)	(D0)	trap number	Allocates a processor trap vector	
FreeTrap	(trapNum)	(D0)	-	Frees a processor trap vector	
CopyMem		(A0,A1,D0)			-624
CopyMem Quick		(A0,A1,D0)			-630
Messages:					
AddPort	(port)	(A1)	-	Adds a message port to the system	
RemPort	(port)	(A1)	-	Removes a message port from the system	
PutMsg	(port,message)	(A0/A1)	-	Puts a message into a message port	
GetMsg	(port)	(A0)	-	Gets a message from a port	-372
ReplyMsg	(message)	(A1)	-	Puts a message to its reply port	
WaitPort	(port)	(A0)	message	Waits for given port to be non empty	
FindPort	(name)	(A1)	port	Finds a given system message port	
Devices:					
AddDevice	(device)	(A1)	-	Adds a device to the system	
RemDevice	(device)	(A1)	-	Removes a device	
OpenDevice	(devName,unit,ioRequest,flags)	(A0,D0/A1,D1)	Error	Gains access to a device	
CloseDevice	(ioRequest)	(A1)	-	Closes a device	
DoIO	(ioRequest)	(A1)	Error	Performs an input/output command and wait for completion	
SendIO	(ioRequest)	(A1)	-	Initiates an input/output command	
CheckIO	(ioRequest)	(A1)	0 or IOReq	Gets input/output request status	
WaitIO	(ioRequest)	(A1)	Error	Waits for input/output request completion	
Resources:					
AddResource	(resource)	(A1)	-	Adds a resource to the system	
RemResource	(resource)	(A1)	-	Removes a resource	
OpenResource	(resName,version)	(A1,D0)	resource	Gains access to a resource	

```
;Your Amiga
;amiga.lib rules ok!
;part 1 -libraries and files
;assembly source code

;External references
;the amiga linker will link these to the program

 XREF  _AbsExecBase     ;exec library address
                        ;always stored in the same location
 XREF  _LVOOpenLibrary  ;exec library routines
 XREF  _LVOAllocMem
 XREF  _LVOCloseLibrary
 XREF  _LVOFreeMem

 XREF  _LVODelay  ;dos library routines
 XREF  _LVOOpen
 XREF  _LVORead
 XREF  _LVOClose
 XREF  _LVOWrite
 XREF  _LVOOutput

main:
 jsr    openlibraries  ;open the libraries
 tst.l  d0
 bne    abort          ;exit if there is an error
 move.l dosbase,a6
 move.l #500,d1
 jsr    _LVODelay(a6)  ;delay a few seconds
 jsr    allocatememory ;allocate a memory buffer
 jsr    readfile       ;read in a text file
 tst.l  d0
 bne    abort          ;exit if the file has not been read
 jsr    filetoconsole  ;copy the file data to the screen
 jsr    freememory     ;free the memory buffer
 jsr    closelibraries ;close the libraries
 rts

openlibraries:
 move.l _AbsExecBase,a6;exec library address
 lea    dosname,a1     ;pointer to string containing dos library
name
 clr.l d0                 ;any version
 jsr   _LVOOpenLibrary(a6);open the dos library
 movem.l d0,dosbase       ;and store the returned pointer
 tst.l d0                 ;check for success
 beq   abort1             ;branch if unsuccessful
 lea   graphicsname,a1  ;open graphics library and store the
pointer
 clr.l d0
 jsr   _LVOOpenLibrary(a6)
 movem.l d0,graphicsbase
 tst.l d0
 beq    abort1
 lea    intuitionname,a1 ;open the intuitionlibrary and store the
pointer
 clr.l d0
 jsr   _LVOOpenLibrary(a6)
 movem.l d0,intuitionbase
 tst.l d0
 beq    abort1
 lea    diskfontname,a1 ;open the diskfont library and store the
pointer
 clr.l d0
 jsr   _LVOOpenLibrary(a6)
 movem.l d0,diskfontbase
 tst.l d0
 beq    abort1
 clr.l  d0
 rts

closelibraries:
 move.l _AbsExecBase,a6     ;close each library in turn
 move.l graphicsbase,a1
 jsr    _LVOCloseLibrary(a6)
 move.l dosbase,a1
 jsr    _LVOCloseLibrary(a6)
 move.l diskfontbase,a1
 jsr    _LVOCloseLibrary(a6)
 move.l intuitionbase,a1
 jsr    _LVOCloseLibrary(a6)
```

▶▶

```
 rts

abort1:                      ;set a flag if there is an error
 move.l  #1,d0
abort:
 rts

allocatememory:
 move.l _AbsExecBase,a6   ;exec library address
 move.l #1024,d0          ;number of bytes required
 move.l #65539,d1         ;type of memory required (chip memory
or clear)
 jsr     _LVOAllocMem(a6) ;allocate the memory
 movem.l d0,buffer        ;store the pointer
 rts

freememory:
 move.l _AbsExecBase,a6  ;exec library address
 move.l #1024,d0         ;number of bytes allocated from this
buffer by

                         ;AllocMem command
 move.l buffer,a1        ;pointer to buffer
 jsr     _LVOFreeMem(a6) ;frees the memory
 rts

readfile:
 move.l dosbase,a6       ;dos library address
 move.l #filename,d1     ;name of file to be read
 move.l #1005,d2         ;the accessmode is 1005 for a file
already open

                        ;1006 to create a new file
clr.l d0
jsr   _LVOOpen(a6)      ;open the file
move.l d0,d1
beq    abort1           ;branch if unsuccessful
movem.l d0,filehandle   ;store the filehandle
move.l buffer,d2        ;pointer to memory buffer
clr.l d0
move.l #1024,d3         ;number of bytes to be read

jsr _LVORead(a6)        ;read the data
cmpi.l #-1,d0
beq    abort1           ;branch if unsuccessful
clr.l d0
move.l filehandle,d1    ;obtain the filehandle
jsr   _LVOClose(a6)     ;close the file
 clr.l d0
  rts

filetoconsole:
 move.l dosbase,a6        ;dos library address
 jsr     _LVOOutput(a6)   ;obtain a filehandle for output to the
console
 move.l  d0,d1
 move.l  buffer,d2        ;pointer to memory buffer
 move.l  #1024,d3         ;number of bytes to be read
 jsr     _LVOWrite(a6)    ;write to the console
 rts

 SECTION  data,DATA
dosname:
 dc.b  'dos.library',0       ;name of dos library
 ds.w   0
graphicsname:
 dc.b  'graphics.library',0  ;name of graphics library
 ds.w   0
intuitionname:
 dc.b  'intuition.library',0  ;name of intuition library
 ds.w   0
diskfontname:                 ;name of diskfont library
 dc.b  'diskfont.library',0
 ds.w   0
filename:
 dc.b  'part1.asm',0         ;name of file to be read, any text
file of more
                             ;than 1024 bytes will do
 ds.w   0
graphicsbase:                ;stored library pointers
 ds.l    1
dosbase:
 ds.l    1
intuitionbase:
 ds.l    1
diskfontbase:

 ds.l    1

 SECTION   mem,BSS
filehandle:                 ;filehandle for file to be read
 ds.l    1
buffer:                     ;pointer to memory buffer
 ds.l    1
```

```
/*Your Amiga*/
/*amiga.lib rules ok!*/
/*part1- libraries and files*/
/*A500 upwards*/

/*c source code*/

long intuitionbase=0;               /*library pointers*/
long graphicsbase=0;
long dosbase=0;
long diskfontbase=0;
long libraries=0;
long buffer=0;                      /*memory buffer pointer*/
long filehandle=0;                  /*filehandle pointer*/
long bytes=0;                       /*bytes read*/
long consolefilehandle=0;           /*output filehandle*/
char *filename="part1.c";           /*any text file will do*/

main()
{
libraries= openlibraries();         /*open the libraries*/
if (libraries == 0) exit(1);
buffer=AllocMem(1024,65539);        /*allocate memory buffer*/
Delay(500);                         /*wait a few seconds*/
bytes=readfile();                   /*read the text file*/
if (bytes ==  0) exit(1);
consolefilehandle=Output();            /*display  it  on  the
screen*/
Write(consolefilehandle,buffer,1024);
FreeMem(buffer,1024);               /*free the buffer*/
closelibraries();                   /*close the libraries*/
}
openlibraries()
{
dosbase= OpenLibrary("dos.library",0);
if (dosbase == 0) return(0);
graphicsbase= OpenLibrary("graphics.library",0);
if (graphicsbase == 0) return(0);
intuitionbase= OpenLibrary("intuition.library",0);
if (intuitionbase == 0) return(0);
diskfontbase= OpenLibrary("diskfont.library",0);
if (diskfontbase == 1) return(0);
return(1);
}
closelibraries()
{
CloseLibrary(dosbase);
CloseLibrary(graphicsbase);
CloseLibrary(intuitionbase);
CloseLibrary(diskfontbase);
}
readfile()
{
filehandle=Open(filename,1005);
if (filehandle == 0) return(0);
bytes=Read(filehandle,buffer,1024);
if (bytes == 0) return(0);
Close(filehandle);
return(1);
}
```

US Action

AMIGA SPECIALISTS

EXTERNAL DISK DRIVES

- Top quality slimline NEC mechanism
- Through port for daisy chaining
- On/off switch
- With or without track display
- 5.25″ drives, 40/80 tracks, IBM/Amiga DOS compatible

3.5″ disk drive	**£99.95**
3.5″ track display disk drive	**£119.95**
3.25″ disk drive	**£129.95**
5.25″ track display disk drive	**£149.95**
3.5/5.25″ multi disk drive	**£229.95**
3.5/5.25″ track display multi drive	**£239.95**
3.5″ internal A2000 disk drive	**£99.95**

RAM EXPANSIONS

A500 512K RAM expansion **£119.95**

- Upgrade to 1 mbyte, fitted in seconds, plugs into expansion port.
- With clock, battery operated to retain time/date.

A500 1.8 mbyte RAM expansion **£149.95**

- Same as A500 specifications except uses 16 × 1 mbyte RAM chips.
- RAM chips not supplied.

A500/1000 2 mbyte RAM expansion **£599.00**

- Fitted in seconds, no soldering, simply plugs into expansion port.
- More space for date storage.
- Faster program execution.

ACCESSORIES

Floppy switch box **£39.95**
- Connects up to 3 drives
- No need to turn your computeron/off to disconnect cables

Clock cartridge **£69.95**
- Fitted in seconds, no soldering
- Battery backed up clock * Retains time and date

A1000 kickstart 1.3/clock **£149.95**
- Plugs into expansion port * New version kickstart
- Battery operated clock

A500 kickstart 1.3/antivirus **£69.95**
- Easy to fit * New version kickstart
- Virus protect program

Boot selector **£121.99**
- Autoboot from an external drive

Eprom programmer **£149.95**
- Easy to use * For A500/1000/2000 (state model)
- Programs 2764/27128/27256/27512
- All A/CMOS types * Complete with software

Genlock A500/1000/2000 (state model
- Professional Genlock, easy to use
- Low cost, enabling you to merge computer graphics with live video and record the results on a VCR

HARD DISKS A500/1000/2000 (STATE MODEL)

- Supplied ready to run
- Built in power supply
- Includes cables and software
- Compatible with RAM expansion units
- You can place your monitor on top of the hard disk unit

20 mbyte hard disk	**£499.00**
30 mbyte hard disk	**£599.00**
40 mbyte hard disk	**£699.00**
60 mbyte hard disk	**£799.00**

MUSIC

Midi interface **£29.95**
- For A500/1000/2000 (state model)
- Compatible with leading packages
- Midi in, 3 midi out, midi thru

Sound digitizer **£39.95**
- For A500/1000 (state model)
- Compatible with leading packages
- Connects to parallel port
- USL for music and speech digitizing

Professional sound digitizer **£99.95**
- The ultimate stereo sound digitizer makes the others sound like toys
- Compatible with leading packages
- For A500/1000/2000 (state model)
- Connects to parallel port

HOW TO ORDER

BY PHONE
☎
U.K. 01-977 9588
HOLLAND 040-733320

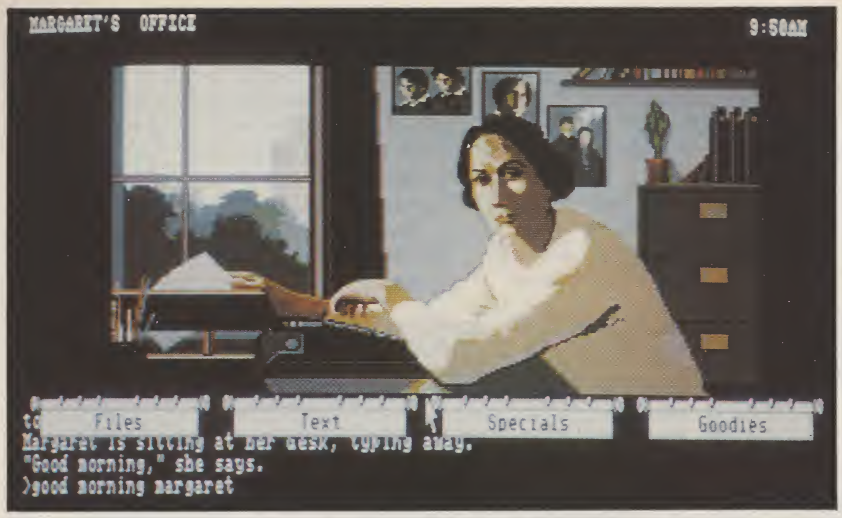
BY POST
Send cheques payable to US Action

BY FAX
U.K. 01-977 5354
HOLLAND 040-433295

U.K. orders FREE
Europe and overseas at Post Office rates
Courier service £6

Address:
US Action, Suite 1,
Wickham House,
2 Upper Teddington Road,
Hampton Wick, Kingston,
Surrey KT1 4DP

US Action, Arote Berg 70, 5611 KL,
Eindhoven, Holland



```
Margaret is sitting at her desk, typing away.
"Good morning," she says.
>good morning margaret
```

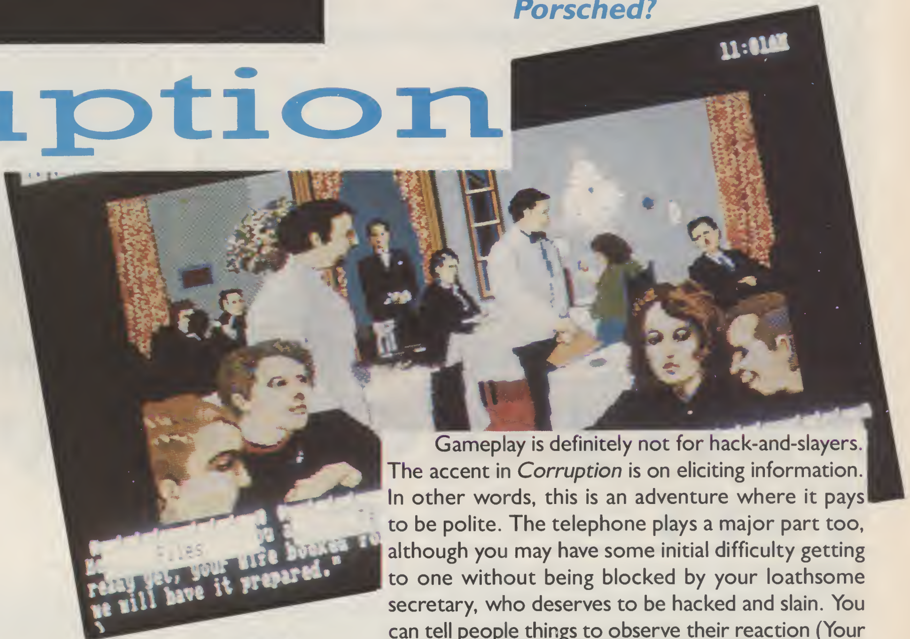
Derek Rogers was top yuppy on a hotly contested ladder. Then, bang, straight in the nick for insider dealing. Did he fall or was he Porsched?

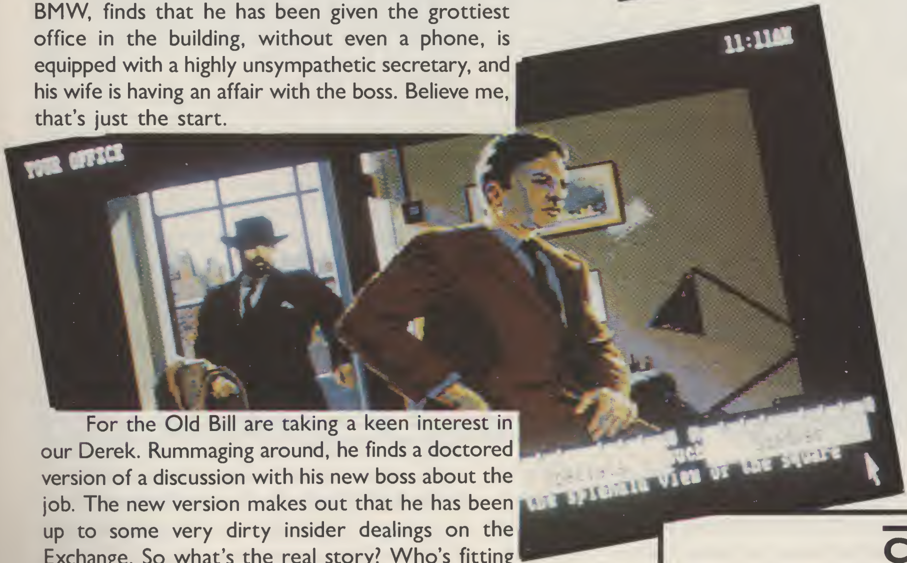
Corruption

One of our writers recently reported that Magnetic Scrolls has just been awarded a British Micro Award for Corruption. I feel that, lest that worthy firm should reach out for its lawyers, I should report that it was actually awarded for *Corruption* and rightly too.

Corruption is a departure from the usual Magnetic Scrolls' subject matter, which in *The Pawn* and *Guild of Thieves* was fantastic, albeit with a wry touch. In this latest game, however, the accent is on present-day realism — well assuming that you think the City is a *real* place.

You play Derek Rogers, a full-time winner, who has just made it to a partnership in a rising electronic firm, due to his winning ways with a portfolio. On Derek's first day he scratches the paintwork on his BMW, finds that he has been given the grottiest office in the building, without even a phone, is equipped with a highly unsympathetic secretary, and his wife is having an affair with the boss. Believe me, that's just the start.





For the Old Bill are taking a keen interest in our Derek. Rummaging around, he finds a doctored version of a discussion with his new boss about the job. The new version makes out that he has been up to some very dirty insider dealings on the Exchange. So what's the real story? Who's fitting him up and why? Most importantly, how can he escape the rap?

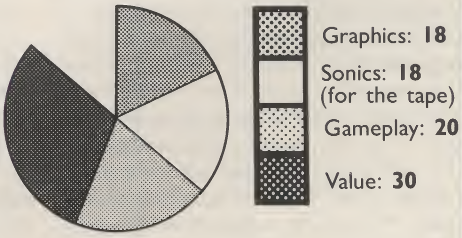
Corruption displays Magnetic Scrolls' usual meticulous care with presentation. Besides the free £500 poker chip, the supplied manual is set on personal organiser pages(for effect, not utility). In it, there are a host of clues, some of which, of course, may be red herrings, together with the usual Hint sequences to type in and some nice humour in the form of some spoof *Time Out* dining out pages. The box also contains an audio cassette which carries an undoctored version of Derek's interview on one side, with the new incriminating version on the other. More clues, doubtless.

Gameplay is definitely not for hack-and-slayers. The accent in *Corruption* is on eliciting information. In other words, this is an adventure where it pays to be polite. The telephone plays a major part too, although you may have some initial difficulty getting to one without being blocked by your loathsome secretary, who deserves to be hacked and slain. You can tell people things to observe their reaction (Your boss is a cool one — faced with the evidence of his affair with Jenny, your wife, he makes a brilliant recovery) and you can also ask them about objects and people. The result of the accent on human beings is that, although you may not visit a lot of locations to start with, there's an awful lot to do in them.

My only problem with the game, and it must be a personal one, is that I found the general atmosphere of the game so cloying and claustrophobic(Derek Rogers after all, is but a rat in a large and complicated trap) that I had to pack it in before I decided to take a trip up to Suicide Bridge. This is not a criticism — indeed it's a tribute to the game's success. I'll probably return to *Corruption* some time soon. In the mean time, I've had to resort to large doses of *Ultima IV* meditation as therapy. OK, yah? ***YA***

CORRUPTION

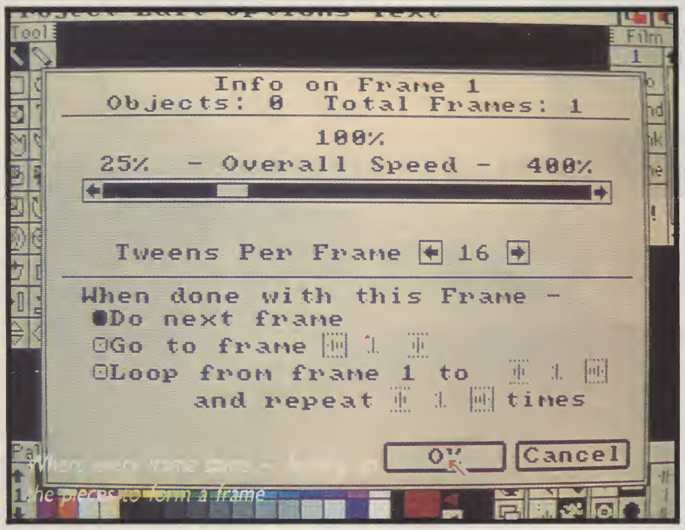
Title: **Corruption**
Supplier: **Magnetic Scrolls/Rainbird**
64-67 New Oxford Street
London WC1A 1PS
Tel: **01-379 6755**
Price: **£24.95**



Improve your image

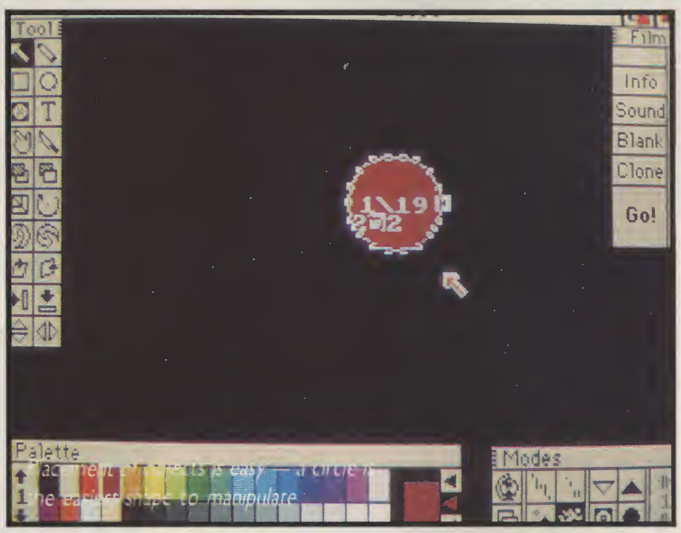
Karen Young looks at two video production packages, both quite different.

The Amiga's superior sound and graphics capabilities have never really been more popular than in the field of video production — not on a large scale I would admit, but more and more small American video studios are incorporating computers in the video studio, and a great many of these are Amiga systems.



Info on Frame 1
Objects: 0 Total Frames: 1
100%
25% - Overall Speed - 400%
Tweens Per Frame 16
When done with this Frame -
● Do next frame
○ Go to frame 1
○ Loop from frame 1 to 1
and repeat 1 times
OK Cancel

When each frame [illegible] the pieces to form a frame



Placement of objects is easy — a circle is the easiest shape to manipulate

This issue I will be looking at two packages for the Amiga. One has no requirement other than a very basic Amiga set-up (A500, 512k RAM), whereas the other requires a good deal more.

Deluxe Productions is yet another member of the *Deluxe* series of software packages, an integrated family of music- and image-manipulating processors that all link in together to form a complete image-making system with some quite advanced music-making facilities to boot!

Deluxe Productions could be viewed as more of a companion package to *Deluxe Video* (*Your Amiga*, September 1988), there are a great many similarities between the two, and where *Deluxe Video* was more suited to designing small "movies" with set pattern of special effects — *Deluxe Productions* is geared towards supplying a better range of wipes, fades and other video effectorama.

The *Deluxe Productions* software is superbly packaged; there are four separate disks, the usual registration card (how many of those have I filled in now?) and the traditional square-format, spiral-bound manual. A nice package and initially it feels as if you are getting value for money.

The disks consist of the program disk, the utilities as well as two art disks containing pictures and images that you will use in your own productions. There is the facility to use standard IFF format images from other software packages — including *Comic Setter* and *Deluxe Paint II*, but more of that later.

First thing is to make a backup copy of the software — there is a hell of a lot of disk accessing going on with this package, and I really recommend making a second security copy in addition to the one Electronic Arts suggest in the manual. You will need an additional drive (at least one) for general everyday applications, but if you are a professional studio, I really would recommend that you consider buying a hard disk as you do not want to have to mess around with disk swapping and all that. Using RAM disk partially solves the problem, but only if you have more than a megabyte — four meg of fast chip RAM will solve the problem nicely.

There are many example productions on the *Deluxe Productions* art disk, as you may have imagined, *Deluxe Productions* uses the standard Intuition interface so common on all of the other *Deluxe* packages.

Rolling your own

One of the most basic components of making a film is to create a storyboard — this is an artist's representation of how the scene will appear when it is finally on the screen (be it video or cinema); *Deluxe Productions* works in exactly the same way, and it is a good example of how basic production edits are created.

A production is broken down into the following components:

Each production can contain up to 12 scenes (each scene can contain up to five edits with one background).

Each edit (also known as a "Clip") can contain an "object" and its "object path" which is a set of points that are set up by you, the user, that describe the movement of the object against the background picture.

Common to all video production studios, a scene and a clip contains "wipes". Wipes control the way images appear on the screen (or alternatively, the way they *disappear* from it as well). These clips must have a "Wipe on" and a "Wipe off", these can be at different speeds and, if it is only a "partial wipe", they can appear only on part of the screen — otherwise, you can have "object wipes" whereby an overlaid image can be faded in and out.

On the production screen, the scene that makes up the current production is shown in "scene boxes". Scenes are normally played in order so that they appear in the production screen, and this is quite easily achieved by placing events "sequentially" after each other in the main screen — this is done, logically enough, by dragging each scene from either a disk file or from other boxes.

A scene can be easily copied this way and, as we shall see later, it can save a lot of time when developing productions from the simplest cartoons and animated diagrams, to the most complex graphic images possible using the 4096 colour HAM (hold and modify) mode.

The Clipboard

If you look at the upper right of the production screen, you will see the Clipboard. This is a form of temporary storage which can be used to store a number of different images to be used at a later date — the more memory you have, the larger the clipboard. It is also handy as a form of "garbage storage" and, as a simple form of scene switcher — a handy item and, as one video producer said to me once, "I wish we could do that with video images, instead of having to fast forward and rewind all the time!" — one day, one day.

Most production screens are built up of three screens; I have already described the Clipboard, but there are, in fact two other screens. The Scene Screen and the Clip Screen.

The Scene Screen is used for editing the components of a scene, and the Clip Screen is used for editing the components of a clip. The Scene Screen appears when you open one of the scenes in the production screen and, likewise, the same occurs with the Clip Screen.

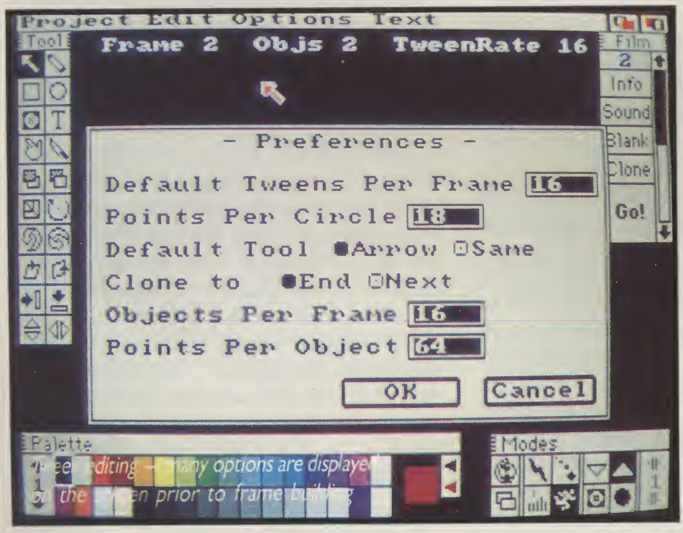
When we get to the Clip Screen, we are really getting into the guts of the system as it displays all of the co-ordinates for each point in the object path. These co-ordinates mark the horizontal and vertical positions of a point, whereas the Object Pauses display the length of time the object stops at each point (in seconds).

The third, and possibly the most vital, piece of information is the Speed gadget — this returns the speed value of an object whilst its companion — the time gadget shows how long it will take the object to move to the next point.

The process of plotting in an object's path is to change the values in the co-ordinates' gadgets — quite frankly, this method is a pain in the butt(and it is a process favoured by other packages not yet available in England). To make things easier, *Deluxe Productions* has an alternative input method by which the package lets you see the object *over* the background. You then move the object to the points you want in its path.

To see the results of your changes to a scene, you select the Play Clip gadget from the menu item. To see the changes, the Play Production icon is clicked on, and this plays the

This is achieved by using the left button on the mouse to select points, and the right button to fix the points in place (thus returning you to the clip screen displaying the figures you *could* have manually entered) — naturally you can edit each number now it is in the grid.



Frame 2 Objs 2 TweenRate 16

- Preferences -

Default Tweens Per Frame 16

Points Per Circle 18

Default Tool ● Arrow ○ Same

Clone to ● End ○ Next

Objects Per Frame 16

Points Per Object 64

OK Cancel

Tween editing — many options are displayed on the screen prior to frame building

A DPAINT 2 graphic imported into Deluxe Productions

changed scene through.

Naturally all the usual "Quit" functions are implemented!

VCR Mode

VCR Mode makes *Deluxe Productions* behave in much the same way as a Video Cassette Recorder would. Like a VCR, all motion and pause funcions are present; they play back the way you set them when you first set up the production and, like a VCR, you can stop and start at any point, and, in addition, you can actually *reverse* your production (play backwards). VCR mode is also very useful if you want to generate stand-alone productions that can be distributed among your friends or colleagues.

Slideshow mode is a little different; you supply the timings between clips (very much like the remote control of a carousel slide show). *Deluxe Productions* pauses at the end of each clip and scene of each production and waits for a click from the mouse for the next clip. Slideshow mode is very useful if you are speaking during playback and want to pace the production with your words.

If one of the picture or object files on your disk is corrupted or missing, the play through on your disk will get to the point where the error occurs and display a system message right in the middle of the screen (or, ir the disk is impossible to validate — in the top left-hand corner of the screen).

Naturally this sort of error would be horrendous if you got an error message on a live broadcast or a production!. To avoid this embarrassing situation, *Deluxe Productions* has two options: The first is an option to test the existence of files required on the disk, whereas the second option tests for the integrity of these files. This latter not only verifies the whole saved productions disk, but also checks to see the files are complete and uncorrupted.

The assistant director should know what is coming up in a production so she can advance from clip to clip at just the right moment. She can use the "Print Production Script" to print out a production script. This script includes the most important information list in a reasonably readable format. No two studios production script forms are the same, and *Deluxe Production's* forms are readable enough, although I would have liked to have seen a customise option for the output whereby the information to be printed out can be altered — much like a forms designer, but I don't suppose it is too difficult to copy out the sheets prior to the production and broadcast meeting.

Using Deluxe Paint II

As I mentioned earlier, there are two different parts to *Deluxe Productions*; backgrounds and objects. Both of these can be imported from other packages and art packages as IFF format files, so Deluxe Paint II is ideally suited to this sort of application, as is Photon Paint and other packages for backgrounds only.

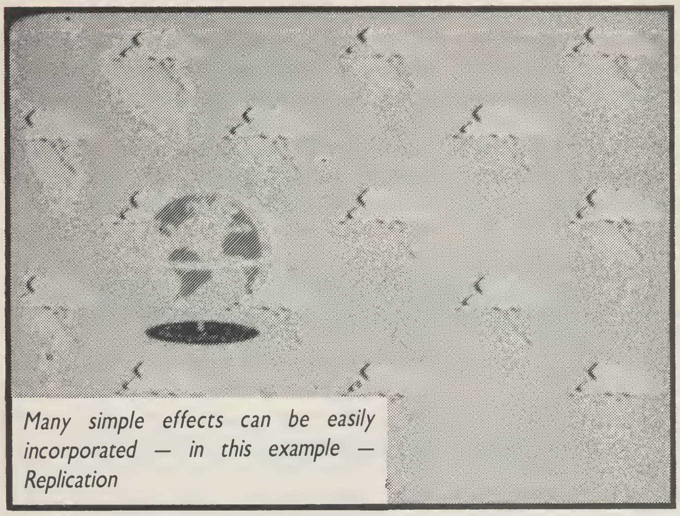
Deluxe Paint II brushes can be used as object in your *Deluxe Productions* production. Thankfully other art packages use similar methods and file formats for the brushes (namely *Comic Setter* and *DigiPaint*) so with a little care, you can use just about any art package you want with *Deluxe Productions*. Expect weird things to happen, though, if you do use any package other than *Deluxe Paint II* as it crashes inexplicably when using large brushes — something to do with *Deluxe Productions*' brush buffer I suppose. Some very high-resolution or very low-resolution brushes will need to be converted by using a package like Butcher prior to use.

So after your initial playing around with the software using pre-drawn images and setting them down in sequence, you will more than likely get to grips with the software using your own images, or "lifting" them from othern packages — thankfully there are so many image manipulation packages around these days that can twist what you have borrowed into something untraceable! Using *Deluxe Productions* I must admit that it took a while before I actually started enjoying using this package, it seemed quite difficult to use initially. For a while it seemed like it was becoming a little tiresome working with *Deluxe Paint II* to create sequential images to be overlaid as mere backgrounds and objects but, after a while things clicked (as they normally do with the *Deluxe* family of programs) and I was able to start creating reasonable presentation images.

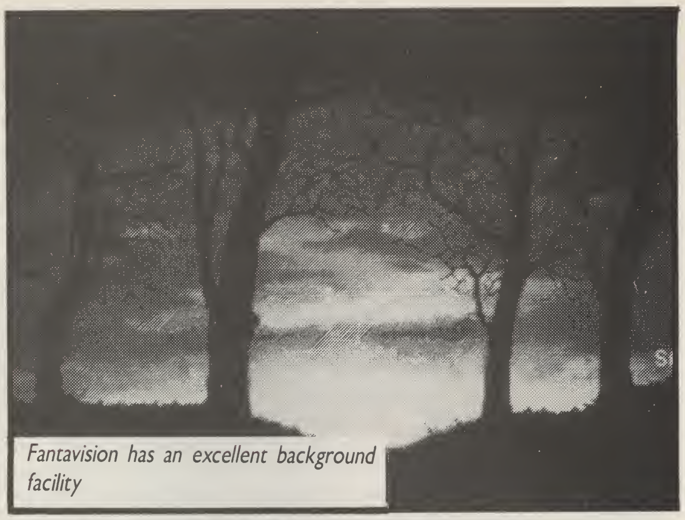
As a video production tool, I strongly urge you to have a look at *Deluxe Video* instead, (review appeared in as it is geared towards producing better and smoother images but, it doesn't have as many good fades and wipes as *Deluxe Productions*. There are 40 in all in *Deluxe Productions* and I hear there is a facility for designing your own wipes coming along soon, this I look forward to seeing if it ever comes to light.

I wasn't suprised to see that the authors of *Deluxe Video* were also the authors of *Deluxe Productions*! There are a great deal of similarities between the two, but I expect this package will find more use at home with people wanting to add a little spice to their home video productions — with the addition of a Genlock you need not have to fade and wipe from one computer image to another, it is equally permissible to use these effects between two different video images!

All in all, *Deluxe Productions* is a competent package, capable of doing a lot



Many simple effects can be easily incorporated — in this example — Replication



Fantavision has an excellent background facility

when some effort is put into making a presentation or a video production — it took me the best part of three weeks before I started creating images approaching anything as smooth and uncluttered as *Deluxe Productions*' examples. You have been warned.

Fantavision

On the other hand, Broderbund software — no stranger to the world of C64 programs — has produced a delightfully simple animation package needing no more than a standard Amiga 500 (no memory expansions needed), although memory expansion can be effectively utilized as a RAM disk if needs be.

You do not need an additional disk drive (as is more or less a necessity with *Deluxe Productions*) and it is this "grass roots" approach to creating videos that makes *Fantavision* so much more easy to use than *Deluxe Productions*. How does *Fantavision* work?

If you have ever done one of those join-the-dots cartoons in childrens' activity books you will probably already have a good idea about how *Fantavision* works; using the mouse, you simply join the dots as a series of points on the screen and the package connects the points to form a representation of solid objects.

This is all well and good if you are "drawing" a picture, but *Fantavision* is an *animation* package, so you create a second frame — probably based on the first but with some changes (no matter how small or large they may be). From these two images, *Fantavision* will do all the laborious work of transforming the shape of an image into the other image — all you have to do is supply the two images.

This transformation process is called 'Tweening'. Up to 128 different Tweens can be produced by the package so transformations can really be smooth — you don't even have to know how the computations involved are calculated, all you have to do is draw.

Basic Elements

Most of the work is done on a window with the palette displayed on the lower left hand screen while the tools you use to draw with are displayed on the whole of the left hand side. There are also a number of modes displayed on the bottom right hand side of the screen — each of these will be described in a moment.

Perhaps the most important window is the Film window — these are the commands you issue to the software enabling sounds to be incorporated into the movie, or allowing you to clone specific sections of the film.

One of the most basic things you can do with *Fantavision* is to create a Flying Object. These objects can be from a simple geometric shape (such as a circle or a square) to a complete image taken from another package using a paint program. The images can be either wire-frame outlines or, as is more likely, a solid colour. It is possible to have outlines *and* "fill colours" as separate entities so they can be changed at any point — this is quite useful when defining flying objects that, for some reason or another, have to move from one coloured background to another, parhaps a ball moving from a green grassy background to a blue sky.

Movie Making

The film window is your guide to where you are in the current production. The number of the frame appears on the top of the screen followed by the other boxes: Info, Sound, Blank and Clone. Changing frames is easily made possible by scrolling from one number to another by using either the scroll bar or, by clicking the arrow to the top and bottom of the bar. And you can insert an new frame by clicking on the "Blank" box. This will insert a new frame either at the end of the movie, or, if you are "filling in" frames — something I will get onto in a minute, then it will work on the new frame created between the current frame and the next one. You can make a copy of a frame easily enough by using the Clone Frame box — this places a copy of the current frame and places it at the end of the movie. Cloning is useful when you have a complicated frame you do not want to redraw.

All About Objects

You build an object in *Fantavision* by plotting a series of points. *Fantavision* keeps track of all your objects and numbers them as they are selected. *Fantavision*'s default setting allows you to have up to 16 objects in any one frame with up to 64 points per object. If you try to go over the limit, *Fantavision* informs you that the frame is full. You may change the configuration of objects and points within the preferences command — the number of objects and points per frame is limited only by available memory, it will make use of additional megabyte memory expansions.

The tools of the trade are many and varied, certainly the fact that they are displayed on the screen all at once, makes it easy to use from the point of view of speedy animation. You use the drawing and editing tools to create and edit objects on the screen. To use them, all you have to do is click the mouse button, and drag their effects on the screen. Draw: This item is the simplest effect and is used to draw an object on the screen. A line that acts like a rubber band appears between the first and the second points, this is how the dot-to-dot animation building blocks are created.

Pointer: This is the basic *Fantavision* tool that all other tools will revert to upon their sucessful (or otherwise) completion. Make Rectangle: Creates a simple rectangle shape, likewise Make Circle creates circles, but also it can create diamonds and other polygons by simply stating the number of sides that a circle must have (four will make a perfect diamond). *Fantavision* can accept standard IFF format images from other packages and other sources (for example digitized images converted to IFF status using a software tool like Butcher). The "Create Bitmap" tool allows you to use only one object number for whatever graphics are enclosed by the icon's wire frame "net".

Fantavision also allows you to import very high resolution images into itself and turn themselves into bitmaps - this is a once only operation, and once selected, you cannot edit them in the same way as you can a standard object — still, you very rarely have to edit things like backgrounds once they are cut and stuck down in place.

Text can also be used in *Fantavision*, although there are a number of limitations over how it can be manipulated on-screen, and how they can be displayed.

All of the work is done inside a "Text Box" and once inside, you can rotate, lean (shear), squash, zoom and flip the same as you would with any other object, but animating text is a little different. You can most certainly move text around the screen, but animating text is a little different than animating other objects, however, text will move around the screen smoothly, but it does not have all of the properties of other tweenable objects. Although the text box can change shape and size itself when tweening, the text it contains will retain its original characteristics. A useful tool is the "front to back and back to front" icon. This is most essential when being used to create any form of distance and perspective as you can set up an order of hierarchy whereby some object appears behind others if they are put on the move.

When you send an object from front to back, it goes all the way to the bottom layer, no matter how many objects are in front of it. The same is true of sending an object to the front. When a group of objects is selected and then the Send from front to back command is used, the entire group will be affected.

There are numerous tools to zoom in, turn (and rotate) images, to lean and squash objects as well as all of the usual copy and padding operations. Each of these options is all very useful but, the one thing that really lets itself down, is in the sound department — a shame, as I feel that sound is unnecessary with such a package — it being best to rely on files created by other packages, for instance, *Deluxe Music Construction Set* or *Aegis Sonix*.

The sound tool when called up produces a box with five sliders and an equal number of boxes, from this box you can load a number of IFF compatible sounds and play them either as a loop or, as a sequence. The way this is achieved is by writing a note for each frame of your movie. This is very cumbersome for music, although it is ideal for sound samples supplied by any 8SVX-IFF file generating sampler such as Datel's or Triangle's. Some of the example files use some sounds that can be pulled into your application (movie) although there are plenty of good demos that can be used — for example sampling sections of minimalist music.

Working with Colour

Fantavision comes with 32 palettes of 32 colours for a total of 1184 different variations for your immediate use. You can modify these colours under the options menu as well as change screen mode.

I must admit that I was a little suprised to see the Amiga's different screen modes — including HAM (Hold And Modify) mode and overscan — quite a feat for Broderbund as HAM mode is by far the hardest one to support as it is so memory intensive. My Amiga 1000 kept crashing on HAM mode operations although the office Amiga 500 and the Amiga 2000 worked perfectly — one can only assume that Kickstart is to blame here.

Fantavision supports other screen modes, such as 2 colour mode, 4,8,16,32 and, as I said earlier, HAM.

A resolution of 672 x 444 enhances *Fantavision* graphics, but operates somewhat more slowly than the lower resolution of 320 x 200, all that is up to you is to choose which one you require. If you need video playback, I would always go for the higher resolution, and cut down the number of moving objects.

You can change palettes at any point throughout the movie, but a new palette is loaded every time you load in a new movie. This is done automatically, so if you fiddle around with the palette during filmmaking, you will have your palette saved, otherwise, you will have to enter the movie in edit mode and then save the whole lot again — a simple operation.

Global Operations

Once you have finished creating your movie, there is still time to make changes. You can make changes to an object in a frame and then automatically apply those changes to the object in all other frames in which it appears. Such overall changes are referred to as "global". You can globally change the colour, animation style or dimension of any single object.

From time to time, you may want to use the screen as an overlay as you create additional frames. You can use an overlay to keep certain images visible as you work on different frames.

You can also do something my animator friends were drooling over for quite a while until they mastered it — set the animation style.

The Animation style you choose determines how *Fantavision* draws the tweens between the frames that make up the movie. *Fantavision* offers four basic animation modes: Normal, Background, Lightning and Trace.

Normal mode is the default setting that you are dropped into as you start working. This is the mode whereby each chance to an object is made frame by frame, and the old object is discarded and erased from the screen.

An ideal example is of a man walking. You supply, say, seven frames of a man walking and choose Normal mode to change each object each time a new frame is loaded in, so that you get a smooth animation effect with no flickering and no after image. Background mode is a little different; you can make an object part of the background. Unlike an overlay, which you use only while you work, a background plays throughout the movie. You can create a background at the beginning of your movie, or you can drop objects into the background temporarily so you can use their object numbers for a new animated object.

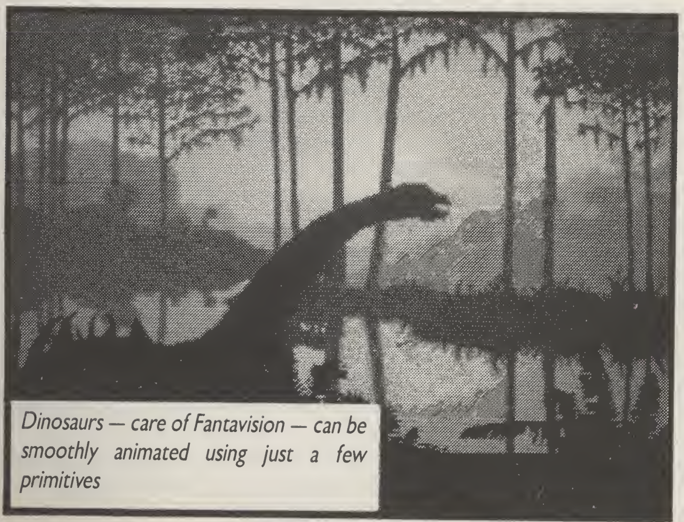
Lightning mode is really a special effect whereby special flashing effects can be set up — like the wonderful ones you see in just about every issue of *Scooby Doo* — with experimentation, you can get really good results if you put an outline around what you want to stand out. The last mode is called Trace mode — this one simply leaves a copy of itself behind it as it moves, very useful if you want to give the impression of speed as you can easily get a character or object to run off the screen in trace mode and then fade out, you can the fade in again with something new, perhaps a different scene, the *suggestion* is that the character has run a long way, (especially if you get your object panting!), it is these sorts of animation tricks that will get you a long way with *Fantavision* and hopefully overcome the natural limitations of the Amiga (when compared to industrial colouring machines and a whole bay of cell painters, artists and inkers).

Conclusions

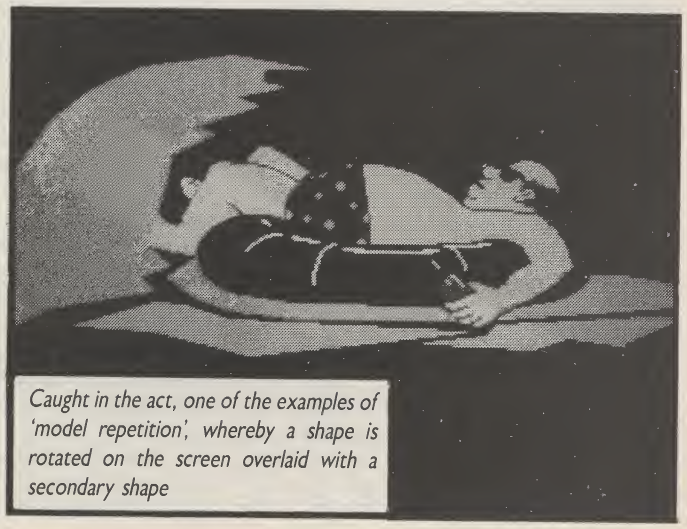
I especially liked this package for its simple, no-nonsense animation and effects. I felt that, for all its technical limitations, I was able to achieve more using *Fantavision* than with any animation package I have yet seen on the market.

Looking at the way the software is presented, I vastly preferred the reasonably cluttered — yet compact — menu system of *Fantavision* to, say, *Deluxe Video*, and I felt that animation response time was vastly improved over that of Electronic Arts' more professional and studied approach to Video making software.

I will not compare *Deluxe Productions* with *Fantavision* as I see *Deluxe Productions* as more of an extension to *Deluxe Video* — what the authors thought would be useful if you needed a less video-orientated package (especially with the slideshow) — The overscan modes are useful, but I believe that they will not be used by a majority of users as very few have professional video recording equipment that will make the most of overscan. *YA*



Dinosaurs — care of Fantavision — can be smoothly animated using just a few primitives



Caught in the act, one of the examples of 'model repetition', whereby a shape is rotated on the screen overlaid with a secondary shape

Corrupt Practices

Amiga disks are notoriously fragile. Burghard-Henry Lehmann explains why disk errors, however, need not be terminal

■ **Amiga disks can easily be corrupted.** All you have to do is take the disk out of the drive while the green light is still on and before you know it, you've got a hard error, as it is called, on the disk. This results in the disk not being acceptedT any more by the DOS and if you put it into the drive you'll get a system error report.

This, however, does not mean that the whole disk is unusable. Most of the time it means that only one or two blocks of the disk have become corrupted.

To mend a disk like this, CLI on Workbench 1.2 has a facility called 'Diskdoctor'. This tells you on which cylinder the hard error is and patches it up. But as a result you will have lost the file where the error was. All you can salvage with Diskdoctor are the files which have not been corrupted. A better way to repair a corrupted disk is "by hand" with a disk editor.

A disk editor is a program that gives you direct access to the data contained on a disk and lets you change it. With most disk editors you choose a block you want to look at and then the disk editor displays the data contained in this block on the screen. You may then change this data in any way you like and get the disk editor to write it back into the appropriate block on the disk.

A disk editor is also useful if you have deleted a file by mistake. If you notice this in time and nothing new has been written onto the disk you may be able to restore the file.

But for this you have of course to know what you are doing, otherwise you might mess up a disk completely! In this article I'd like to give you the background information which you'll need to access and change a disk successfully. If you haven't got a disk editor, don't despair! In the next issue of *Your Amiga* I'll provide you with one.

The Structure of an Amiga disk

Before you can use a 3.5-inch disk on your Amiga it has to be formatted or initialised by Amiga DOS, the disk operating system of the Amiga. AmigaDOS formats a disk into 80 cylinders, 160 tracks and 1760 blocks.

The 80 cylinders are numbered from 0 to 79. Each cylinder consists of 2 tracks - the so called "top track" on one side of the disk and the "bottom track" on the other side, since all disks used on the Amiga are double-sided. This means that there is a total of 160 tracks on an Amiga disk.

Each track in turn is formatted into 11 sectors, numbered from 0 to 10. This gives a grand total of 1760 blocks (80 * 2 * 11) on each disk, which are numbered from 0 to 1759.

When accessing and changing a disk directly with a disk editor you'll look at a certain block. So it is the blocks that interest us most.

Blocks

Each block holds 512 bytes of data. From this you can calculate that an Amiga disk can hold a total of 901 120 bytes of data, even though only about 520 000 bytes of this contains pure data, the rest is used to manage the disk and the filing system. Incidentally, if you want to make maximum use of the capacity of an Amiga disk, use few or no directories. This saves on filing information and frees therefore more disk space for data proper!

THE BITMAP

Hex	FF	F0	BF	FF
Decimal	255	240	191	255
Binary	11111111	11110000	10111111	11111111
Block	897	880 (Root Block)		866
Block	897	881		866
Binary	11111111	11110000	00111111	11111111
Decimal	255	240	063	255
Hex	FF	F0	3F	FF

This table shows how Block 881 is changed from free to used. To identify blocks correctly, read from right to left.

These 512 bytes are portrayed by most disk editors on the screen in chunks of 4 bytes or "long words", as they are also called. A long word is 32 binary bits long (because of this it is also called a "32-bit number"). Therefore, each block contains 128 long words (512/4).

To explain this a bit further, let's assume a long word is pointing at block 350 of the disk. Most disk editors portray the data contained in a block in hexadecimal because this is more convenient than decimal. Decimal 350 is hex 015E, and since it is shown in form of a long word, block 350 would be portrayed as 0000015E.

Conversely, a long word can hold 4 ASCII characters. For example, long word 6469736B reads as "disk" (hex 64 is ASCII "d", hex 69 is ASCII "i" and so on).

To find a certain position on a block with a minimum of counting, one roughly halves the block and counts both from the beginning and the end of the block, whichever is nearer. For this purpose in layout diagrams like figure 1 to 4 the first long word is given as 0, the second as 1, the third as 2 and soon, while the last long word is given as Size-1, the one before that as Size-2, the one before that as Size-3 and soon.

Since the Amiga disk filing system consists of directories, sub-directories and data files, there are five different types of "filing system blocks", as they are called:

- * The Root Block
- * The User Directory Block
- * The File Header Block
- * The File List Block
- * The Data Block

Furthermore, there are two types of blocks which are of equal importance to the management of a disk, but which are only mentioned in very advanced literature:

- * The Boot Block
- * The Bitmap

The two Boot Blocks, which are always block 0 and 1, are of little interest to you, since they are only used to boot a disk. But the Bitmap, which tells the DOS which sectors on the disk are used and which ones are free, is quite important. That's why we will have a close look at it in a minute.

The Filing System

The layout of the five types of filing system blocks is quite consistent and can therefore easily be grasped:

The first six long words (0 to 5) contain the header information and the checksum. This is also called the primary block identification and it tells you what kind of block you are looking at.

Next there are 72 long words which contain the hash table (more about this next issue). The hash table holds the block numbers of the User Directory Blocks, File Header Blocks and Data Blocks the block is pointing at.

Next there are 27 long words (Size-50 to Size-24) which either contain the Bitmap pages, if it is the Root Block, or the protection flags and the text of the directory or file comment if it is a User Directory Block or a File Header Block.

The next 19 long words (Size-23 to Size-5) which give the creation date of the disk or directory or file and its name.

Finally there are four long words (Size-4 to Size-1) which give the so-called secondary block identification.

The Root Block

The Root Block is, apart from the two Boot Blocks, the only fixed block of a disk. It is always block 880, that is cylinder 40, track 0, sector 0. Thus the Root Block lies exactly in the center of the disk. All the other blocks are installed in a fluid manner as the disk is filled up and its contents are changed.

To find your way around a disk with a disk editor, you first call up the Root Block (most disk editors do this automatically when you pop a new disk into the drive). The Root Block tells you where the first generation User Directory and File Header Blocks are and where the Bitmap is. In turn, the first generation User Directory Blocks tell you where the second generation User Directory and File Header

Figure 1: Layout of the Root Block

0	00000002	Type
1	00000000	Header key (always zero)
2	00000000	Highest sequential number (always zero)
3	00000048	Hash table size (always 72)
4	00000000	Unused
5	nnnnnnnn	Checksum
6	nnnnnnnn	Hash table
Size-51	nnnnnnnn	
Size-50	FFFFFFFF	Bitmapflag (-1 if Bitmap is valid)
Size-49	nnnnnnnn	Pointer to blocks containing the Bitmap
Size-24	nnnnnnnn	
Size-23	DAYS	Date and time when disk has been altered last
Size-22	MINS	
Size-21	TICKS	
Size-20	DISK	Name of disk in ASCII
Size-8	NAME	
Size-7	CRE-DAYS	Date and time when disk has been created
Size-6	CRE-MINS	
Size-5	CRE-TICKS	
Size-4	00000000	unused
Size-3	00000000	unused
Size-2	00000000	unused
Size-1	00000001	Secondary block identification

Figure 2: Layout of User Directory Block

0	00000002	Type
1	OWN KEY	Header key (points to self)
2	00000000	Highest sequential number (always zero)
3	00000048	Hashtable size (always 72)
4	00000000	Unused
5	nnnnnnnn	Checksum
6	nnnnnnnn	Hashtable
Size-51	nnnnnnnn	
Size-50	00000000	unused
Size-48	PROTECT	Protection bits
Size-47	00000000	unused
Size-46	nnnnnnnn	Comment string stored in ASCII
Size-24	nnnnnnnn	
Size-23	DAYS	Creation date and time
Size-22	MINS	
Size-21	TICKS	
Size-20	DIRECTORY	Name of directory in ASCII
Size-5	NAME	
Size-4	nnnnnnnn	Hashchain pointer (zero if hashchain finished)
Size-3	nnnnnnnn	Back pointer to parent directory
Size-2	00000000	unused
Size-1	00000002	Secondary block identification

Figure 3: Layout of File Header Block

0	00000002	Type
1	OWN KEY	Header key (points to self)
2	nnnnnnnn	Total number of data-blocks in file
3	nnnnnnnn	Number of data block slots used
4	nnnnnnnn	First data block
5	nnnnnnnn	Checksum
6	DATA B-n	List of data blocks
	DATA B-3	
	DATA B-2	
Size-51	DATA B-1	
Size-50	00000000	unused
Size-48	PROTECT	Protection bits
Size-47	nnnnnnnn	Total size of file in bytes
Size-46	nnnnnnnn	Comment string stored in ASCII
Size-24	nnnnnnnn	
Size-23	DAYS	Creation date and time
Size-22	MINS	
Size-21	TICKS	
Size-20	FILE	Name of file in ASCII
Size-5	NAME	
Size-4	nnnnnnnn	Hash chain pointer (zero if hash chain finished)
Size-3	nnnnnnnn	Back pointer to parent directory
Size-2	nnnnnnnn	Pointer to first extension block (zero if none)
Size-1	FFFFFFFD	Secondary block identification

Figure 4: Layout of File List Block

0	00000002	Type
1	OWN KEY	Header key (points to self)
2	nnnnnnnn	Number of data blocks in block list
3	nnnnnnnn	Same as above
4	nnnnnnnn	First data block
5	nnnnnnnn	Checksum
6	BLOCK n+n	Extended list of data block keys
	BLOCK n+3	
	BLOCK n+2	
Size-51	BLOCK n+1	
Size-50	00000000	unused
Size-5	00000000	unused
Size-4	00000000	unused
Size-3	nnnnnnnn	Back pointer to File Header Block
Size-2	nnnnnnnn	Pointer to next extension block (zero if none)
Size-1	FFFFFFFD	Secondary block identification

Figure 5: Layout of Data Block

0	00000008	Type
1	OWN KEY	Header key (points to self)
2	nnnnnnnn	Sequential number
3	nnnnnnnn	Size of data
4	nnnnnnnn	Next data block
5	nnnnnnnn	Checksum
6	nnnnnnnn	DATA

Blocks are, and, finally, the File Header Blocks point you to the File List Blocks and the Data Blocks which contain the data itself which is stored on the disk.

Let's now look a bit closer at the layout of the Root Block (see figure 1): The first long word gives you the block type number which is 2. (This is the same for the User Directory Blocks, the File Header Blocks and the File List Blocks. Only the Data Blocks have a different block type number, which is 8.) The next two long words (1 and 2) of the Root Block are always zero because they are not used. The fourth long word (3) gives the size of the hash table which is always 72 (hex 48). The sixth long word (5) contains the checksum. (Note that 5 always contains the checksum on any of the filing system blocks!) The checksum is the sum of all the long words in the block, expressed in a negative number. For those of you who, like me, are none the wiser after a statement like this, let me give you an example:

Imagine that all the long words in a block are 00000000, that is the whole block contains zero. In this case the checksum too is 00000000. Now one long word is 00000001. In this case the checksum is FFFFFFFF (decimal -1). If another long word would now contain 00000002 then the checksum would be FFFFFFFD (decimal -3), that is 00000000 — (00000001 + 00000002). The important thing about the checksum of any block is that is has to be correct. If it isn't, as happens frequently when a block has become corrupted, then the disk is considered to be faulty by the DOS and not accepted. Most disk editors allow you to mend this easily by calculating the correct checksum for you.

The Hash table

The next 72 long words contain the hash table, that is the block numbers of the User Directory Blocks and File Header Blocks the Root Block is pointing at.

It is called the hash table because the entries contained in it are sorted in a special way which is called "hashing" or "applying a hash function".

Whenever you have a large number of data which you want the computer to search you can employ different methods of sorting that data. The best known way with alphanumeric or string data is of course sorting it alphabetically, like in a dictionary. In this case the "hash function" (or the sorting algorithm) which you apply works like this: All words starting with the letter 'a' are given the hash code 1, all words starting with 'b' are given the hash code 2, and soon. Hash code 1 is of course the first position in the hash table, while hash code 2 is the second position.

Now all words which start with the same letter and therefore yield the same hash code are put into what is called a "hash chain". That is, each word in the chain has a pointer which points to the next word in the chain. If there are no more words in this chain, the pointer contains zero, meaning that this particular hash chain is finished.

The idea of hashing is of course to come up with a hash function which makes the search through the data as efficient and speedy as possible. This means, you want to prevent "bunching", where most data is contained in one or two hash chains which are very long.

Unfortunately I haven't got the information as to which hash function AmigaDOS uses to chain directories and files. It certainly doesn't do it alphabetically! I will, hopefully, be able to enlighten you next issue.

Nevertheless, the directories and files contained in a disk are chained in the following way: Let's say position 10 of the hash table of the Root Block points at block 883. This might be a User Directory or a File Header and it is the first link in the hash chain with hash code 10. To find the next link in that hash chain you look at Size-4 of that User Directory or File Header Block. If this contains zero then the chain ends with this block. Otherwise the hash chain continues with the block whose number is given in Size-4.

Therefore, when Amiga-DOS searches for a particular directory or file it first uses the hash function on the name to extract the hash code. Then it searches through all the names in that hash chain. If a name matches with the name of the directory or file it is searching for it has found the User Directory Block or File Header Block of that directory or file. If it comes to the end of the hash chain, that means that the search has been unsuccessful.

Next the Root Block tells you where the Bitmap is and if it is valid.

If the Bitmap is valid, Size-50 (remember, this is counted backwards from the end of the block!) contains FFFFFFFF (that is -1). Size-49 to Size-24 contains the Bitmap pages, even though I haven't seen a disk yet which contains more than one Bitmap.

Next Size-23 to Size-21 contains the time, that is, the days, the minutes and the ticks, when the disk has been altered last. After this (Size-20 to Size-8) comes the name of the disk given in ASCII. Then the original creation date of the disk is given (Size-7 to Size-5).

Size-4 to Size-2 is not used on the Root Block and therefore contains zero.

Finally, Size-1 or the last long word of the Root Block contains the secondary block type number which is 1.

The User Directory

If there are any directories or subdirectories on the disk then there is a User Directory Block for every one of these.

As you can see from figure 2 the layout of the User Directory Block is rather similar to the layout of the Root Block.

The main differences are in the bottom of the block: Size-4 contains now the pointer to the next link in the hash chain (or zero, if the hash chain is finished).

Size-3 points back to the parent directory or the Root Block.

The File Header Block

The File Header Block (see figure 3) heads a file which is stored on the disk. If it is a large file with more than 72 Data Blocks then there is a pointer to a File List Block which serves as an extension of the File Header Block. Otherwise the Data Blocks are listed where the hashtable is on the Root Block and the User Directory Block.

If there is an extension, Size-2 points to the first File List Block.

The secondary block identification number of a File Header Block (as well as a File List Block) is FFFFFFFD (decimal -2).

The File List Block

As I've said above, a File List Block (see figure 4) is employed if a file is larger than 72 Data Blocks. If one File List Block is not sufficient to point to the rest of the Data Block there is a further File List Block whose block number is given in Size-2, the extension pointer.

The Data Block

Finally we come to the Data Block (see figure 5) which contains the data stored in a file itself.

The Data Block uses only the first six long words for filing information. The rest of the block is free for the data proper. Therefore, each Data Block can contain up to 488 bytes of data.

To chain the Data Blocks of a file in the right order, all Data Blocks are given a sequential number starting from 1. This number is contained in position 2.

The Bitmap

In order to store new data on a disk the DOS has to know which blocks are used and which ones are free. This information is stored in a special block, called the Bitmap.

The Bitmap really is very simple: Each block on the disk is represented by a binary digit. If the bit is on (=1) then the block it represents is free. If the bit is off (=0) then the block is used.

The first long word of the Bitmap contains the checksum. Then the Bitmap itself starts. But it starts with block 2, since block 0 and 1 are always used as the two Boot Blocks (as a matter of fact, blocks 0 and 1 are appended at the end of the Bitmap). Thus the second long word contains the allocation information for block 2 to 33, while the third long word contains the information for block 34 to 65 and soon.

To find a block on the Bitmap you first have to find the long word or position where the block is included. To do this subtract 2 from the block, divide the number of the block by 32 and round the result up to its nearest integer; if the result is an integer, add 1. For example, 2-2/32=0, 0+1 gives position 1; 34-2/32=1, 1+1 gives position 2; 880-2/32=27.43, rounded up to nearest integer gives position 28 and soon.

As a matter of fact, figure 6 is based on a disk which has been newly initialised on Workbench. Naturally, most of the blocks on such a disk are free, which means that most long words contain FFFFFFFF. Only in the center part of the disk five blocks are used (the Root Block, the Trashcan Directory) and the Trashcan Information.

Figure 6 shows you how you can change a block from free to used. In this example we do this with block 881. The hex number which contains this block is BF. This converts into binary 10111111, showing block 181, which is represented by the first binary digit, as free. To allocate this block the binary number becomes 00111111 and this converts into hex 3F. Therefore to allocate block 881 you have to change the hex byte BF to 3F.

In the next article I will give you the promised disk editor and demonstrate in more practical ways how you can use it. *YA*

Guru's Guide

Another month, and the Guru comes down to answer some more of your technical questions.

Thank you for the many letters sent to me last month, It was indeed a great surprise to find so many people writing in to me. Naturally there are is only a finite space for me to place my answers in, so I shall get down to the nitty gritty of answering them now.

Memory Management

Dear Guru,
First of all, congratulations of a super column, I found your answers regarding the Paula Chip in last month's *Your Amiga* to be truly "enlightening".

My problem concerns the supposed "incompatibility" between Amiga 1000/500 programs and the 1 Megabyte Amiga 2000 programs that are fast becoming more and more popular!

Admittedly these problems mainly occur when attempting to play arcade games, but I have also noticed a number of quirks when using WordPerfect (strange characters appearing in my documents, and spurious crashes when printing, or when accessing a disc drive for a long time).

I replaced my Amiga 1000's memory upgrade thinking it was at fault, but the problem only arises when I attempt to run specific games (Star Wars being the most popular of the "incompatible" ones) and when I attempt to load some multi-tasking applications, I get a guru message — are you trying to tell me something?

I would be grateful if you could tell me what is going wrong. It would seem a shame to have to remove the 512K memory upgrade just to run certain applications or games, any suggestions?

Ms L Morgan, Bengeo, Herts

Quite a few Ms Morgan, one of which is to carefully inspect the pins of your memory expansion.

You see, it is quite possible for some memory expansions to operate with only, say, three quarters efficiency (750K) simply because a pin at the end of the memory rail is either broken or not connecting correctly, if you look also at the connector, ensure that the connections on the Amiga's main board are secure and that the DIL socket is not wobbly!

Believe it or not, the easiest way to check for this is to load up the workbench disc, make sure that there isn't a Ram disc set up or whatever, and simply boot up the disc at the workbench prompt, if you get a figure around the 390000 bytes mark then you are running under a 512K memory map. If you have a figure closer to 9500000 bytes then you are running a full megabyte.
Don't be worried if you cannot account for, say 100K or so missing from the memory map, a great deal of memory is taken up by the workbench, and a similarly worrying amount is taken up by the RAM Disc.

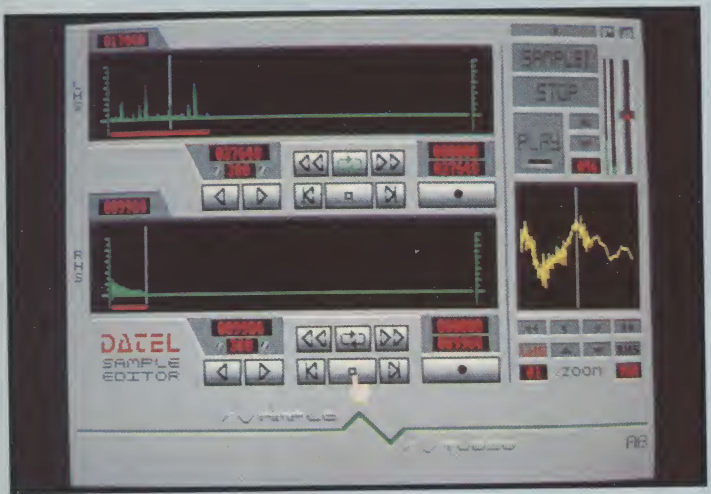
If we have ascertained that the fault lies not in the board or the Amiga, but the software you are running, then don't worry, there is a package that resets the Amiga to a Workbench prompt, and gives you just the standard 512K memory map.

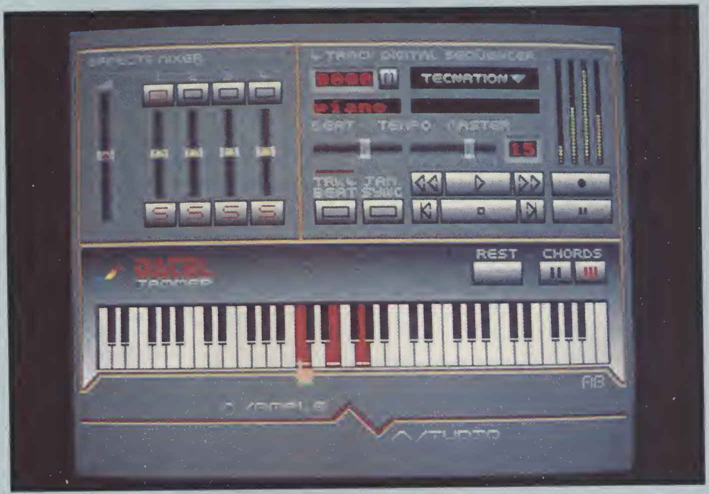
This item came into the office last month and it is called INT-SWITCH, and is is effectively a disc one puts onto your machine before you boot off your hitherto unbootable discs, it comes with the Guru's seal of approval, I use it myself, and it is very clever about the way it works, you can try to access the memory by using some sort of memory monitor, but unless you know how to enter the right sequence of binary "blips" the 68000 processor gives to Fat Angus — the memory manager chip (amongst other things) at startup, then the chances are, this package will make your upgrade invisible to the operating system.

The only problem I can see with this package is the possible confusion that can arise when booting off of a disc with auto parameter settings (such as weird memory manager discs on the PD) — these discs use illegal calls to the Amiga's EXEC and thus put the

Continued Page 56▶

DATEL ELECTRONICS





☐ AMIGA PRO SAMPLER STUDIO + DATEL JAMMER

- A top quality sound sampling system at a realistic price.
- 100% machine code software for realtime functions.
- HiRes sample editing.
- Realtime frequency display.
- Realtime level meters.
- Files saved in IFF format.
- Adjustable manual/automatic record trig level.
- Variable sample rate & playback speed.
- Seperate scroll line waveform windows plus zoom function with Edit windows for fine accurate editing.
- 3D shot of sound waveform. Wave editor to design your own waveforms or adjust existing ones.
- Microphone & line input 1/4" Jack & Din connections.
- Software files can be used within other music utilities.

ONLY £69.99 PLEASE STATE A500/1000/2000

To complement the Sample Studio the Datel Jammer gives you a 5 octave keyboard to play & record your sampled sounds.

FEATURES:-

- 4 track sequencer up to 9999 events.
- Tempo & Beat controls.
- Mixer Controls on Instruments.
- Load & Save sequence.
- Works on standard IFF file sounds.

☐ MIDIMASTER

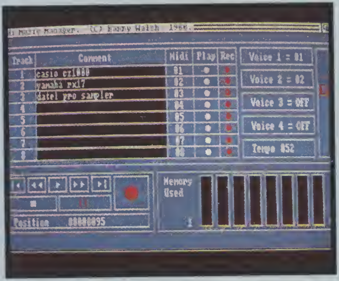
- Full Midi Interface for A500/1000/2000 (please state model).
- Compatible with most leading Midi packages (including D/Music).
- Midi In - Midi Out x3 - Midi Thru.
- Fully Opto isolated.
- No need to pay more - Full Midi standard.

ONLY £34.99

☐ MIDI CABLES

- Top quality.
- 3 metre length.

ONLY £6.99 PAIR
UNBEATABLE VALUE



Music Manager. (C) Danny Walsh 1988.

Track	Comment	Midi	Play	Rec
1	casio cz1000	01	●	●
2	yamaha rx17	02	●	●
3	datel pro sampler	03	●	●
4		04	●	●
5		05	●	●
6		06	●	●
7		07	●	●
8		08	●	●

Voice 1 = 01
Voice 2 = 02
Voice 3 = OFF
Voice 4 = OFF
Tempo 052
Memory Used
Position 00000095

☐ MIDI MUSIC MANAGER

A TRULY PROFESSIONAL MIDI PACKAGE AT A REALISTIC PRICE

- Play sampled sounds on Amiga from any Midi track.
- Full dubbing - listen to one track while recording another.
- Works with many Midi interfaces including Datel Midi Master (see Ad) .
- 8 realtime Midi tracks for record/playback.
- Adjustable track length - limited only by available memory.
- Works with standard IFF files.

ONLY £39.99

☐ DISC/STORAGE BOX OFFERS

- DD40 holds 40 3.5" discs. Lockable.

ONLY £6.99

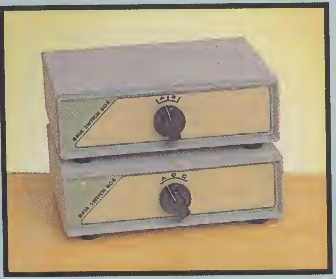
- DD80 holds 80 3.5" discs. Lockable.

ONLY £8.99

☐ DISCS

- 3.5" DS/DD.
- Top quality, bulk packed.

ONLY £22.99
FOR 25 DISCS



☐ DATA/SWITCH BOXES

- A/B type connect two printers to one computer or vice-versa.
- Centronics connections or RS232 Serial connections (25 pin). Please state.

ONLY £24.99

- ABC type connect three printers to one computer or vice-versa.
- Centronics or RS232 connections.

ONLY £34.99

☐ PRINTER LEADS

- 25 pin 'D' to 36 way Centronics parallel lead. 1.2m length.
- A500 or 1000, please state.

ONLY £8.99

☐ ROBOTARM

FULL FUNCTION - 5 AXIS MOVEMENT

- Explore the fascinating science of Robotics with this full feature Robot Arm.
- Human like dexterity - with 5 Axis of movement it is so versatile. It can manipulate small objects with amazing ability.
- Easily controlled using 2 joysticks (any 9 pin type) or connect to your Amiga with our Interface + Software to give Computer/Robotic control (see Interface offer).
- Comes with Accessories including 'Finger' Jaws, Magnetic Attachment, Shovel Scoop, 4 Stabilizing Suction Base Legs, etc.
- Uses 4 HP2 batteries (not supplied) to power motor movement so uses no computer power.
- Self contained, ready to use (except batteries, joysticks).

ONLY £49.99

☐ INTERFACE OFFER

- Unique Software/Hardware package to allow you to interface your Amiga with the Robotarm.
- Train mode allows you to store & then repeat movement sequences.
- Very easy to use.
- This interface is not needed to use the Robotarm but interfacing with your Amiga has great possibilities.

ONLY £24.99
COMPLETE WITH CABLES.

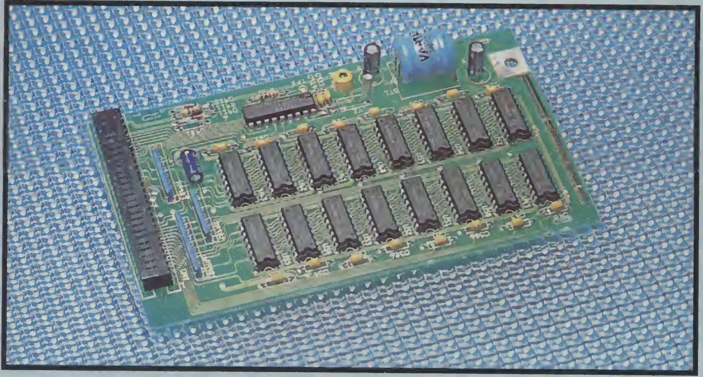
DATEL ELECTRONICS

☐ EXTERNAL 3.5" DISC DRIVE

- Slimline extra low profile unit - only 6" long!
- Top quality NEC drive mechanism.
- Throughport allows daisy-chaining other drives.
- A superbly styled case finished in Amiga colours.
- Fully compatible.
- 1 meg unformatted capacity.
- Good length cable for positioning on your desk etc.
- Value for money - before you buy a drive please compare the features - this drive has an NEC drive unit & is housed in a superb housing - many units available are built to a price & not a standard. Don't spend a few pounds less & end up with 'rubbish' - & remember you are buying from the manufacturer.
- Complete - no more to buy.
- Single or twin drive models available.

ONLY £89.99 SINGLE DRIVE £149.99 TWIN DRIVE

ADD £5 FOR COURIER DELIVERY IF REQUIRED



☐ 512K RAM EXTENSION CARD

- Available with/without calender/clock option.
- Simply plugs internally into A500 slot.
- Switch in/out with switch supplied.
- Fitted in minutes - no soldering etc.
- Accepts 41256 DRams (zero K fitted).
- With calendar/clock onboard time/date automatically booted.
- Battery backed to retain time/date.

ONLY £19.99 FOR STANDARD CARD TO ACCEPT 512K

ONLY £34.99 FOR VERSION WITH CLOCK/CALENDAR

PHONE FOR LATEST FULLY POPULATED BOARD / RAM PRICES.

☐ MARAUDER II

QUITE SIMPLY THE BEST DISC COPIER AVAILABLE FOR THE AMIGA (ALL MODELS)

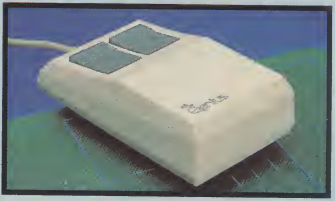
- Superfast disc copier will copy almost any commercial disc.
- Friendly user Interface - Mouse driven throughout.
- Completly compatible with Amiga multitasking system.
- Even decrypts many encoded programs including D. Print/Video/Paint/Music/II etc.
- Supports up to 4 drives simultaneously for multple copies.
- Special 'Strategy Files' cope with even the most advanced protection schemes.
- Now shipping the latest version.
- Fast option - typically around 80 seconds.
- USA's top selling copier.

ONLY £29.99 POST FREE

☐ EXTERNAL DRIVE SWITCH

- Switch in/out of external drives.
- Save on memory allocated for drives not currently in use.
- DF1 & DF2 controlled.

ONLY £9.99



☐ REPLACEMENT MOUSE

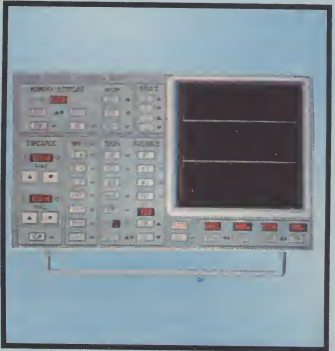
- Fully Amiga compatible.
- Rubber coated ball.
- Optical type.

ONLY £24.99

☐ SPLITTER LEAD

- Allows joystick & mouse to be connected to same port.

ONLY £4.99



☐ DATA ACQUISITION UNIT

- Turn your Amiga into a sophisticated measuring instrument capable of measuring a wide range of data inputs.
- Sample & display events from microseconds to hours- with amplitudes from milivolts to 50 volts.
- A Hardware/Software package with very high spec. including:-

DIGITAL SCOPE DISPLAY - 2 channel inputs.Manual or continuos display. Timebase 500ms/div to 20us/div- accurate to 5%.

- 6 bit flash conversion gives 2 million samples/sec.

PLOTTER DISPLAY

- Timebase range 1 sec to 10hrs per plot.

All features found on units costing thousands of pounds.

ONLY £89.99

PLEASE STATE A500/1000/2000

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0782 744707 24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	0782 744292 UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

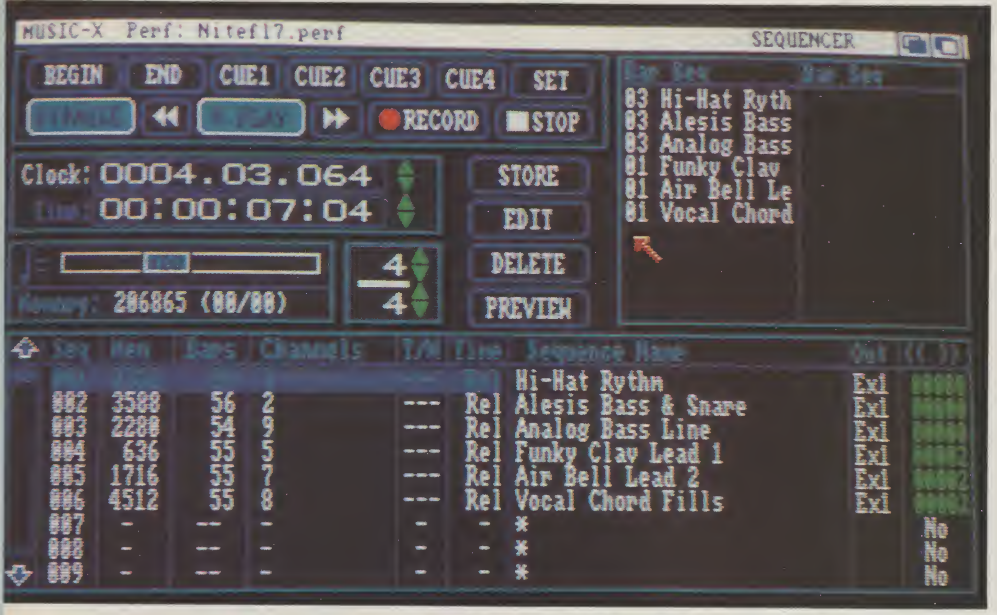
PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLERS WELCOME - Please reserve goods by telephone prior to visit.

DATEL ELECTRONICS

DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE
GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.

SALES ONLY	TECHNICAL ONLY
0782 744707	0782 744324



```
MUSIC-X  Perf: Nitef17.perf                    SEQUENCER
Clock: 0004.03.064
       00:00:07:04
       286865 (00/00)
4/4

03 Hi-Hat Ryth
03 Alesis Bass
03 Analog Bass
01 Funky Clav
01 Air Bell Le
01 Vocal Chord

                                Hi-Hat Rythm             Exl
002  3588   56  2   ---  Rel  Alesis Bass & Snare      Exl
003  2280   54  9   ---  Rel  Analog Bass Line         Exl
004   636   55  5   ---  Rel  Funky Clav Lead 1        Exl
005  1716   55  7   ---  Rel  Air Bell Lead 2          Exl
006  4512   55  8   ---  Rel  Vocal Chord Fills        Exl
007   -     --  -    -    -   *                         No
008   -     --  -    -    -   *                         No
009   -     --  -    -    -   *                         No
```

Music X

If you thought you'd seen the definitive sequencer package for the Amiga just listen to what Darrin Williamson has to say about Micro Illusions' latest offering

■ **There are a lot of different MIDI** packages now available for the Amiga. From sequencers and voice editors to generic patch librarians, all claiming to be "professional" which usually just means difficult to master.

So you can imagine how sceptical I was when I visited the *Mediagenic* and saw another sequencer package billed as a world-beater. "Ho-Hum", I thought. Then I saw it in action...

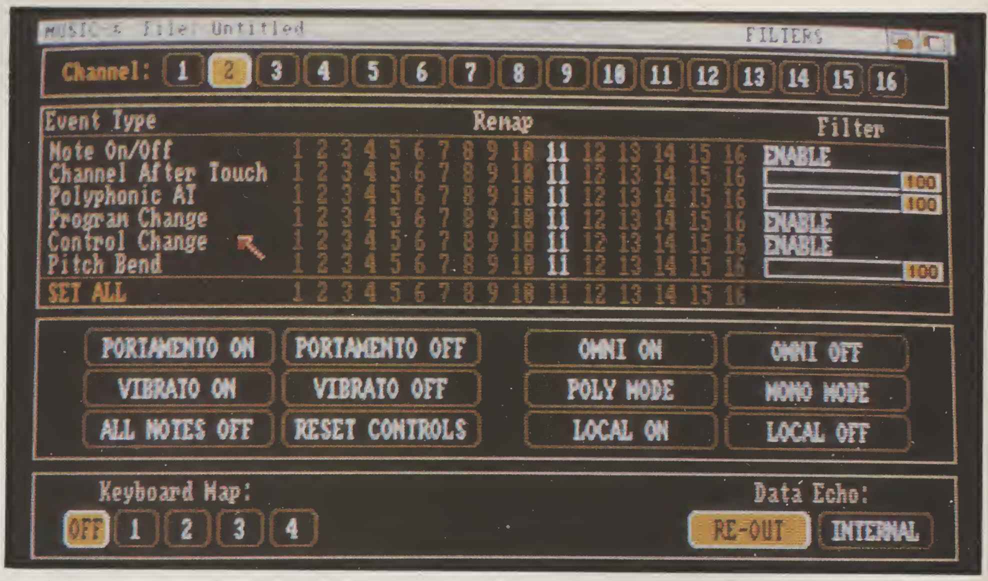
I had to find out more about this package, so, after we all got over PC Show-lag, I blew the entire *Your Amiga* expenses budget for the year on a second class return to Southampton to get an exclusive peek at *Music X* as it approaches completion. Wanna know what it can do? Read on...

How many tracks?

First and foremost *Music X* is a sequencer package, however Micro Illusions sees it as a musicians' tool rather than just a sequencer. Why? Because on top of being a first rate step and realtime MIDI recorder *Music X* is a complete MIDI workstation on disk. I shall, however, leave the juicy bits until later. I'll tell you about the sequencer first.

A bog standard A500 will give you a record/playback capacity of something like 25 000 MIDI events. This is enough for the average length pop song so no worries there. These events can be spread over 250 (that's right, 250) tracks which is over four times more than its nearest competitor. This means that you can really play the indecisive producer and have a dozen different bass lines or even a number of different riffs to choose from. This number would also be useful for synth expander users with a set of drum sounds on one MIDI channel (like the MT-32 for instance). Each drum sound could be recorded onto a different track which would allow an incredible degree of flexibility when chaining your sequences into songs. You aren't limited to external MIDI sounds either, you can store up to 16 internal sound (memory permitting) and play up to four simultaneously. These can be IFF, Sonix or *Music X* format so plenty of scope there.

As you can see the layout of the screen is comprehensive yet uncluttered and makes good use of the Amiga's Desktop environment. In the top left hand corner of the screen are the tape transport controls which act as you would imagine. At the bottom of the screen is your track list which tells you what is on each track, what MIDI channel it's transmitting to and from which bar it will come in on. This particular feature is quite interesting in that you can record a sequence with everything starting at the beginning of the and experiment bringing in instruments one by one which is the way most 12-inch singles are spun out. As you'll notice as I go on, the ability to experiment with the manipulation of MIDI data without doing irreparable damage to your potential chart-topper is a key feature of this package. It brings back the creative curiosity which has been lacking in so many of the music packages I have seen. Throughout the entire program you are subconsciously telling yourself to try out new ideas. This has got to be a good thing for the music industry.



```
MUSIC-X  File: Untitled                              FILTERS
Channel: 1 [2] 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Event Type            Remap                                      Filter
Note On/Off           1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16    ENABLE
Channel After Touch   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16    100
Polyphonic AT         1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16    100
Program Change        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16    ENABLE
Control Change        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16    ENABLE
Pitch Bend            1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16    100
SET ALL               1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Keyboard Map: [OFF] 1 2 3 4                  Data Echo: [RE-OUT] INTERNAL
```

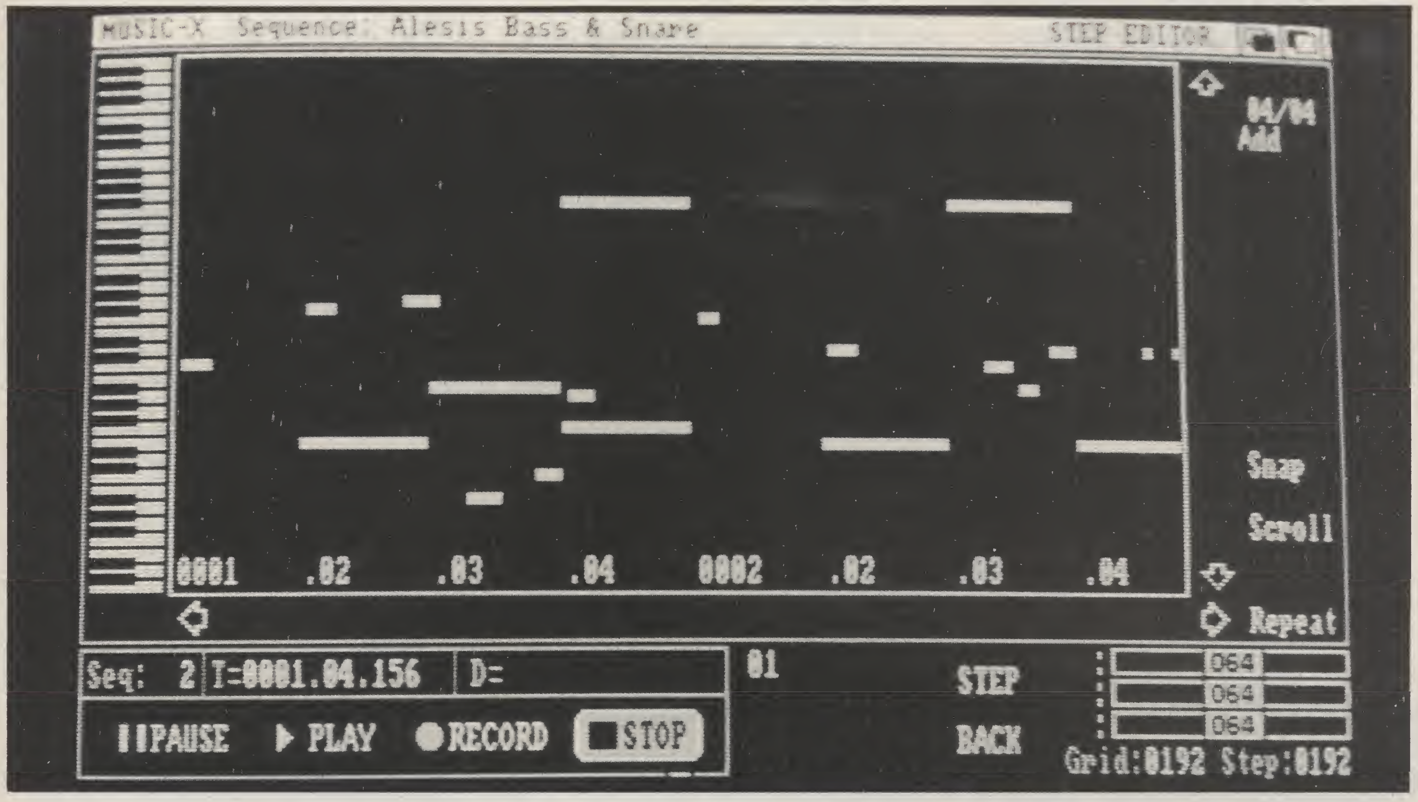
As with multi track tape machine *Music X* has the ability to drop in and out of record mode at any place in the recording, which is great if you just want to redo a small section of a sequence. However unlike a tape machine you can drop in and out precisely down to a fraction of a second thanks to the built-in SMPTE timeclock. Furthermore the point won't drift at all as there is no tape involved to slip or overshoot.

In addition to drop ins, you also have four Cue points which you can set anywhere in you piece of music. This saves a lot of tedious mucking about looking for the start of the middle eight for instance. Once set a click of the mouse on the marker will take you straight to your desired destination.

Another invaluable feature of this particular segment of the program is the MIDI channeliser which allows you to experiment with sending a recorded track to any MIDI channel. This is useful for checking which of your sounds suit a particular sequence. Again, nothing is permanent until you save it.

You can synchronise the sequencer in five different ways. The default is internal, however you can hook it up to the MIDI clock from a drum machine, a SMPTE timecode generator, any form of video locking system or from a MIDI timecoder such as the little Tascam unit on sale for about £120.

The final area of interest on this section of the program in the mixdown area which allows you to mute one, some, or all of the tracks in a sequence allowing a degree of automated mixdown. This area would appear have a lot of potential however it was not quite finished in the version I saw.


MUSIC-X Sequence: Alesis Bass & Snare

STEP EDITOR

04/04 Add

Seq: 2 T=0001.04.156 D=

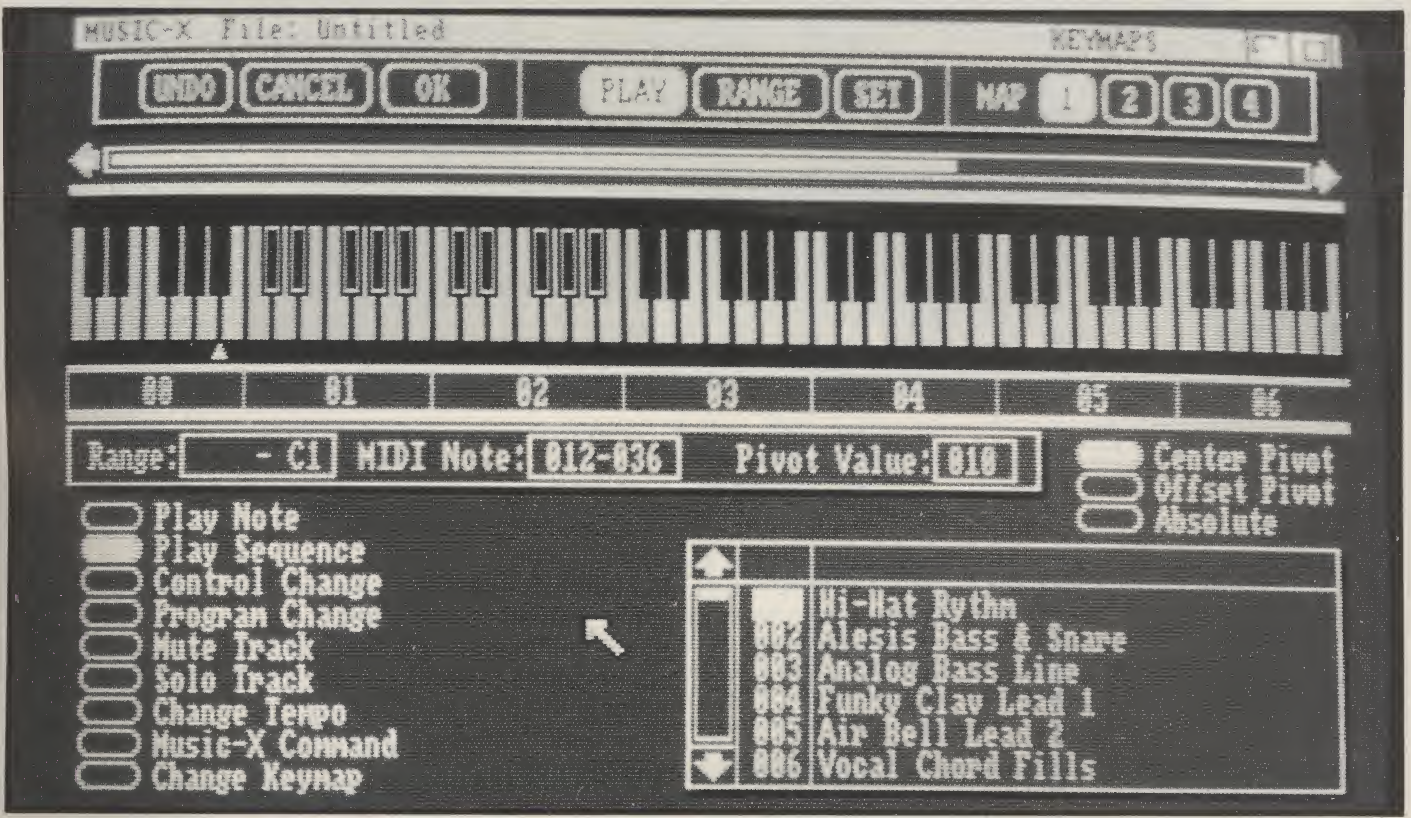
01

Once everything is recorded into *Music X* you will no doubt want to correct a few mistakes you have made or add a few fiddly bits in step time. For this job we have not one but two editors at our disposal.

Firstly we have the Event Editor which lists out, in sequential order all the events on one track as MIDI values. This allows very precise editing of all recorded MIDI information on one track totally independent from all the others. So you could do some very strange things by, for instance, changing the tempo of just one track within a sequence. This may sound like a bad idea on paper but may sound great in a sequence. More immediately useful is the ability to scroll through a sequence slowly in step time and insert, delete or alter specific notes. All sequencers do this to a certain extent but *Music X* is the only one I've come across that actually plays the notes in step time making the hunt for that bum note a hell of a lot easier.

From within this area you can quantise individual tracks. Nothing special in that, except you can quantise all permutations of note start, note duration and note end but you can vary the percentage of quantisation so that on a course setting only really wayward notes get corrected. Ones near the mark are left where they make a sequence sound far more human.

Complementing the Event Editor is the Bar Editor, which does much the same job but graphically. Notes are shown as white horizontal bars alongside their corresponding note on the keyboard drawn out on the left hand side of the screen. The longer the white bar, the longer the duration of the note. The taller the blue vertical bar behind the white one


MUSIC-X File: Untitled

KEYMAPS

Range: - C1 MIDI Note: 012-036 Pivot Value: 010

● Center Pivot
○ Offset Pivot
○ Absolute

○ Play Note
● Play Sequence
○ Control Change
○ Program Change
○ Mute Track
○ Solo Track
○ Change Tempo
○ Music-X Command
○ Change Keymap

Hi-Hat Rythm
002 Alesis Bass & Snare
003 Analog Bass Line
004 Funky Clav Lead 1
005 Air Bell Lead 2
006 Vocal Chord Fills

the louder the velocity level. I Know it sounds a bit confusing to describe it like this, but, take it from me, it's dead simple to pick up once you see and hear it in action. Editing on this screen is simply a matter of clicking the mouse where you want the new or corrected note to be. Once again, you can mess around with things without ruining anything. Presumably some people will find they get on better with the event editor, some will prefer the bar version personally I can see myself flipping between the two for different things.

Those three elements alone would constitute a very nice package but that's only the tip of the iceberg as we now see what MIDI utilities *Music X* has to offer.

MIDI Magic

Once you get into MIDI and begin acquiring drum machines, synths, expanders and so on, you soon need some convenient way of swapping your MIDI IN, OUT and THRU-puts, as you don't really want to keep yanking out 5-pin DIN leads and shoving them in elsewhere, you want something to do that for you. Several Hardware manufacturers supply MIDI patchbays, the cheapest being Akai's at £199. With *Music X* a full 16X16 MIDI patchbay comes as just another section of the main program.

It's no ordinary patchbay either. with this one you can not only send the input of one channel to the output of another but you can also split the incoming data up and, for instance send the note information to one channel but the aftertouch data to another. Again, this sounds weird, but it could be useful and certainly makes creative use of the MIDI standard. In this area you can also vary the percentage of effect that data has on another channel. The sound you have on channel one may sound great with a liberal smattering of modulation, however the sound on channel two may be more appropriate but may sound awful with that level of modulation. So to get over this problem you just take down the percentage of the effect that data has on that channel using the on-screen sliders.

Next we have a powerful keymap utility which basically allows you to split up your keyboard so that different keys send MIDI messages to different things. This is useful if you have a bass sound on channel one and a piano sound on channel two. You can split your keyboard so that the bass sound is triggered from the lowest octave and the piano from the rest of the keyboard. Far more complicated set-ups are possible which can involve a different sound/program changes on each key. This is not only useful for people in studios (home or commercial) with their master keyboards but also those who wish to perform live and control their synth rack from a remote keyboard. I think this is the first time a computer sequencer package has been invented which actually helps live music.

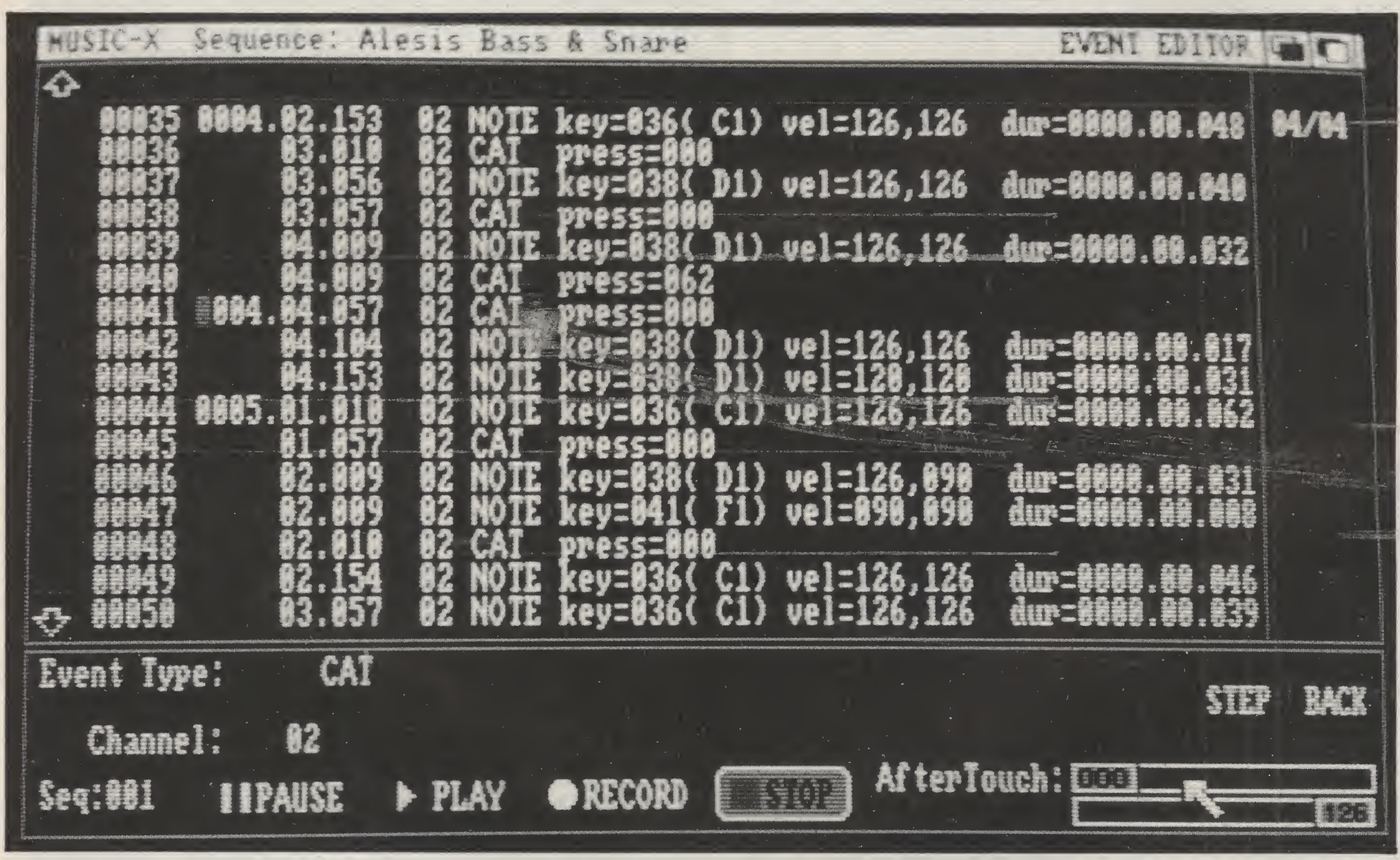
One further feature of this portion of the prog is the fact that it is also a MIDI monitor. Yet another £200 piece of rack-mounted hardware you don't need to buy!

Finally, the Librarian. This is a generic patch librarian which can store all the system exclusive voice and performance data of your synths, drum machines and so on. This is a godsend for three reasons. First, doing MIDI data dumps onto Amiga disks is a lot quicker and more reliable than saving to tape. Second, it allows you quick and easy access to groups of sounds which can be saved together with the songs to which they belong, so that you can load in both the song and all the synth voices needed for that song to the corresponding units in one fell swoop. Thirdly, the cheapest program that does this on its own costs nearly 100 and the cheapest hardware equivalent costs over £300.

A large number of download protocols already appear on the disk for the most popular instruments. There are, however, plans to produce additional data disks for use with this area of *Music X* containing new download protocols and possibly voice data as well.

What more can I say? *Music X* is not only a damn good sequencer but also a comprehensive set of MIDI tools to really make your system (big or small), sing. It takes full advantage of all the Amiga's facilities thanks to skillful programming from a true Amiga programmer. This package was written exclusively for the Amiga, and it shows. As proof it even runs as a background task. In short, if Dr T's KCS was the Rolls Royce of Amiga sequencers (as Mary Branscombe stated in her review last issue) then *Music X* has got to be the *Porsche 959*.

If that doesn't make my feelings on *Music X* clear then how about this to stick on Micro Illusions' advertising copy "*Music X* is without doubt the best piece of MIDI software to date. Buy it!".



```
MUSIC-X  Sequence: Alesis Bass & Snare                        EVENT EDITOR

 00035 0004.02.153  02 NOTE key=036( C1) vel=126,126  dur=0000.00.048  04/04
 00036        03.010  02 CAT  press=000
 00037        03.056  02 NOTE key=038( D1) vel=126,126  dur=0000.00.040
 00038        03.057  02 CAT  press=000
 00039        04.009  02 NOTE key=038( D1) vel=126,126  dur=0000.00.032
 00040        04.009  02 CAT  press=062
 00041 0004.04.057  02 CAT  press=000
 00042        04.104  02 NOTE key=038( D1) vel=126,126  dur=0000.00.017
 00043        04.153  02 NOTE key=038( D1) vel=120,120  dur=0000.00.031
 00044 0005.01.010  02 NOTE key=036( C1) vel=126,126  dur=0000.00.062
 00045        01.057  02 CAT  press=000
 00046        02.009  02 NOTE key=038( D1) vel=126,090  dur=0000.00.031
 00047        02.009  02 NOTE key=041( F1) vel=090,090  dur=0000.00.000
 00048        02.010  02 CAT  press=000
 00049        02.154  02 NOTE key=036( C1) vel=126,126  dur=0000.00.046
 00050        03.057  02 NOTE key=036( C1) vel=126,126  dur=0000.00.039

Event Type:    CAT
                                                         STEP  BACK
  Channel:   02

Seq:001   PAUSE   PLAY   RECORD   STOP   AfterTouch: 000
                                                              126
```

Next issue I should have my paws on one of the first production copies of *Music X* in the country. I'll be reporting on just how well it fares in my home studio by taking you through, step by step, the process of writing a song on a computer. *YA*

Title: **Music X**
Supplier: **Mediagenic**
23 Pond Street, Hampstead, London
NW3 2PN
Tel: 01-431 1101
Price: **£195 (To be confirmed)**

To find out all you need to

Know

about the Amiga computer

it

pays to subscribe to 'Your Amiga'

all

you need to do is fill in the coupon

U.K. £9.00, Europe £11.80, Middle East £11.90,
Far East £13.00, Rest of the World £12.10
Airmail rates on request.

Please commmence my subscription to YOUR AMIGA
with the .... issue. I enclose my
cheque/money order for £................ made payable to
ARGUS SPECIALIST PUBLICATIONS LTD.
or debit £................. from my Access/Barclaycard
No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
Valid from .... to ....
Signature ....
Name ....
Address ....
.... Postcode ....

Send this form with your remittance to:
INFONET LTD. 5 River Park Estate,
Billet Lane, Berkhamsted, Herts HP4 1HL

AMIGA CENTRE SCOTLAND

4 HART STREET LANE, EDINBURGH. EH1 3RN
Tel. 031-557-4242 : FAX 031-557-3260

MAKE THE MOST OF YOUR AMIGA WITH THE LATEST SOFTWARE AND HARDWARE RELEASES

KIND WORDS KIND WORDS

THE FIRST FULL FEATURE WORD PROCESSOR FOR THE AMIGA COSTING UNDER 50 POUNDS FEATURES INCLUDE HAM GRAPHICS, SPELLING CHECKER, MAILMERGE AND SUPERFONTS FOR HIGH RESOLUTION PRINTING.

ANNOUNCING THE AMAZING HANDYSCANNER

AT LAST AN INEXPENSIVE AND EFFECTIVE SCANNING DEVICE FOR THE AMIGA

SCANNED IMAGES

UK REGISTRATION AND PRODUCT SUPPORT FOR MOST OF OUR PRODUCTS.
SOFTWARE FULL PAL VERSIONS
(Where available)

PLEASE SEND LARGE STAMPED ADDRESSED ENVELOPE FOR A COPY OF OUR LATEST BROCHURE.

THE COUNTRIES' LEADING AMIGA SOFTWARE SPECIALISTS

SOFTViLLE PD SERVICE

55 HIGHFIELD AVENUE, WATERLOOVILLE
HAMPSHIRE PO7 7PY, ENGLAND. Tel 0705-266509

o SMALL selection from our collection of over 400 disks FULL of the best PD available for your

AMIGA

SOF7 - "PROBE", the best demo of its type.
SOF2 - TUTORIAL, crommed with information
SOF9 - PERFECT SOUND ployer/recorder etc.
SOF15 - CLIP ART in IFF formaot
SOF20 - NASA digiview slideshow, 14 pics
SOF26,27 - Two disks full of Comms progroms
SOF10 - FULL of MIDI progrms
SOF11 - DISK TWO of MIDI utilities
SOF17 - RAY TRACING construction set
SOF21 - Pagesetter CLIP-ART
SOF25 - JET FLIGHT dema from Sublogic
SOF33,34,35 & 36 - Four disks full of the very best demos from the Badge Killer contest
SOF40 - The fabulous "Walking Legs" dema
SOF43 - AmigaSpell, spelling checker
SOF45,46,47 & 48 - Four disks full of tunes for use with the SONIX progrom.
SOF49 - FONTS & FONT UTILITIES
SOF50 - Printer Driver Generotor
SOF54 & 55 - The brilliant NEWTEK demo. You need both disks, two drives and 1Meg
SOF56 - The incredible "DOC demo + 4 others
SOF57 - WORDWRIGHT Word processor
SOF59 - JUKEBOX MUSIC demo, great samples
SOF60 - Digitised music, over 11.5 hrs worth!
SOF66 - FIVE sampled tunes - holf megs only
SOF67 - Boatbench V2.0 - make your own demo programs you've olways marveled over.
SOF68 - CLASSIC BOARD GAMES
SOF69 - CLASSIC ARCADE/BOARD GAMES
SOF70 - CLASSIC AMIGA DEMOS
SOF72 - Four great grophic mini-movies
SOF73,74,75 & 76 - Miami Vice, digitised. You need oll four disks (but only one drive).

All these plus the full range of FRED FISH disks (currently up to FF154), the best of the AMICUS collection, all the FAUG disks, the complete set of PoNorAmA disks ond the SLIPPED DISK collection. Whatever you want, we've got it.

NO MEMBERSHIP - NOTHING TO JOIN

We simply provide the very BEST service to enable you to get great software cheaply. Plus GUARANTEED same day postage of all orders.

Disk prices; £3:00p each - £2:75p for 6 & over - £2:50p for 11 & over
Deduct £1:00p per disk price if you send in your own blank disks.
Cheques/po's payable to 'SOFTVILLE' at above address.

We also stock a great range of essential goods for every Amiga owner
PRO-5000 J/Stick (clear or black) - £13:50 Quickshot2 (Turbo) - £11:00
Disk Boxes (lockable) to hold 50 - £8:00 to hold 100 - £10:00
Library case to hold 10 - £1:50 MOUSE PADS, good quality - £3:50
CUMANA 1meg DRIVES - £104:99
A full range of Amiga specific books at discount prices - ask for list
Commercial Software at 25% off r.r.p
e.g. Great Giana Sisters - Fire & Forget - £18:25 each
plus Labels, Paper, Blank Disks, branded and bulk etc. etc.

PUBLIC DOMAIN SOFTWARE

Send a stamp for our FREE 24 page catalogue, state clearly "AMIGA" and this magazine.

BRITAINS No1 PD SUPPLIER

Assembly Language Programming

Peter Lawrence sets out to quench your thirst for 16-bit programming with another heady draught of 68000 code

■ **In this installment we'll be looking at** a variation upon the example presented in the previous issue. Once again we'll be changing the background colour but in this case using a custom screen rather than the workbench. Instead of using boolean gadgets to select preset colours this example sets up proportional gadgets and allows any of the Amiga's 4096 colours to be selected.

Before presenting the example program let's first take a look at the new structures, constants and routines that are needed.

Local customs

To open a custom screen it is first necessary to set up and initialise a *NewScreen* structure. This structure is defined as follows:

```
WORD    ns__LeftEdge, ns__TopEdge, ns__Width      {0,2,4}
WORD    ns__Height, ns__Depth                     {6,8}
UBYTE   ns__DetailPen, ns__Block__Pen             {10,11}
UWORD   ns__ViewModes, ns__Type                   {12,14}
APTR    ns__Font, ns__DefaultTitle                {16,20}
APTR    ns__Gadgets, ns__CustomBitMap             {24,28}
```

LeftEdge — Not currently used, since Intuition only supports custom screens which are as wide as the entire video display. It should, however, always be set to zero to allow compatibility with future releases which may allow for variable screen widths.

TopEdge — The video line number at which the screen top should be.

Width — Set to either 320 (LoRes) or 640 (HiRes).

Height — The height, in video lines. Due to a limitation in the system software *TopEdge* and *Height* should never be set such that the bottom of the screen is above the bottom line of the video display.

Depth — The number of bit planes to be used for the screen.

ViewModes — These flags determine the screen's characteristics:

HIRES (=$8000):	Set if the screen is HiRes.
INTERLACE (=4):	Set if the screen is interlaced (400 lines high).
SPRITES (=$4000):	Set if sprites are to be used on the screen.
HAM (=$800):	Sets the screen up in Hold and Modify mode.
DUALPF (=$400):	Sets Dual Playfield mode.

HAM and *DUALPF* are advanced Amiga modes, unfortunately beyond the scope of this treatment.

Type — This should be set to *CUSTOMSCREEN*(=$F)

Font — A Pointer to the *TextAttr* structure for the font to be used in the screen. If this is set to zero then the default font will be used.

DefaultTitle — Pointer to a *NULL* terminated text string which will be used in the screen's title bar.

Gadgets — *Screen* may contain gadgets (they do not have to be attached to windows) and this field should contain a pointer to the first gadget in a list. If there are no gadgets then this field should contain zero.

CustomBitMap — It is desirable under certain circumstances for the application program to set up the bitmap memory itself. If this is the case then this field should point to the bitmap and the Type field should be set to *CUSTOMBITMAP* (=$40).

To set up a proportional gadget it is necessary to supply more information than is contained in the *Gadget* structure. The *SpecialInfo* field should be set to point to a *PropInfo* structure which is defined below. The *Gadget* rendering specified in the main *Gadget* structure refers to the rendering of the proportional gadgets knob.

```
UWORD   pi__Flags, pi__HorizPot, pi__VertPot     {0,2,4}
UWORD   pi__HorizBody, pi__VertBody              {6,8}
UWORD   pi__CWidth, pi__CHeight                  {10,12}
UWORD   pi__HPotRes, pi__VPotRes                 {14,16}
UWORD   pi__LeftBorder, pi__TopBorder            {18,20}
```

Flags:

FREEHORIZ(=2) and *FREEVERT* (=4): when set, allow the gadget to move in the horizontal and vertical planes respectively.

PROPBORDERLESS(=8): causes Intuition to draw the gadget without a border around it.

KNOBHIT(=$100): This flag is set by Intuition whenever the user selects the knob, as opposed to selecting any other part of the gadget. This flag can be tested by the application program.

HorizPot, VertPot — These variables hold the value contained by the gadget as an unsigned 16-bit number. Initial values should be set by the program.

HorizBody, VertBody — These variables contain the increment by which the pot can change. Selecting the gadget above or below the knob will cause the pot value to increase or decrease by this much.

CWidth, CHeight, HPotRes, VPotRes, LeftBorder, TopBorder — Set and maintained by Intuition.

In the example program the knob of each gadget is rendered as an image and so we also need to define an *Image*structure.

```
WORD    ig__LeftEdge, ig__TopEdge               {0,2}
WORD    ig__Width, ig__Height, ig__Depth        {4,6,8}
APTR    ig__ImageData                           {10}
UBYTE   ig__PlanePick, ig__PlaneOnOff           {14,15}
APTR    ig__NextImage                           {16}
```

LeftEdge, TopEdge — Location of the image within the gadget.
Width, Height — Size of the image in pixels.
Depth — Number of bit planes in the image, which may be less than or equal to the depth of the screen.

ImageData — This is a pointer to the image data in the form of a series of 16-bit words. Each bit represents a pixel within a bit plane starting with the top, left-hand corner of the image one bit plane at a time. If the image width is not an exact multiple of 16 then extra bits are ignored.
PlanePick — Each bit of this field represents one of the screen's bit planes. Setting these bits determines which bit planes the image will be loaded into. If the screen has 4 bit planes and the image 2 then if *PlanePick* is 3 the image will be loaded into the first two bit planes. If *PlanePick* is 5 then the image is loaded into the first and third bit planes.
PlaneOnOff — This field specifies what to do with the other bit planes not directly effected by the image. If the corresponding bit is not set then the bit plane will be filled with zeros; if it is set then the bit plane will be loaded with ones.
NextImage — If more than one image is to be rendered then this field should point to the next *Image* structure. If there are no more images then set this to zero.

Library Routines

This example uses two new routines from the Intuition
library.
OpenScreen {-198} — This routine assumes that register A0 contains a pointer to a *NewScreen* structure and uses this data to open a custom screen. To maintain the screen Intuition sets up a *Screen* structure and when the *OpenScreen* routine returns it leaves a pointer to this *Screen* structure in register D0.
CloseScreen {-66} - When a custom screen is finished with it should be closed to remove it from the display and free up the memory it used. When this routine is called A0 must contain a pointer to the relevant *Screen* structure (note: not the *NewScreen* structure). This function makes no allowance for any windows which may still be open within the screen.

68000 Instructions

In the example program we also introduce a new 68000 instruction, the LSR (Logical Shift Right). This is one of a series of shift and rotate instructions which form part of the 68000's instruction set. Basically, shift and rotate instructions both involve moving each bit within a *BYTE*, *WORD* or *LONGWORD* one place, either to the right or the left.
The end bit which is moved out of the data element is moved into the carry bit of the status register.

It the other end of the data element (for example, the right end in a left shift operation) the last bit has no adjacent bit from which to receive a new value so the problem is what should be placed into this bit. The answer to this question defines the difference between the shift and the rotate instructions.

In a rotate instruction the final bit is filled from the carry bit of the status register. Since this is where the first bit is moved to the bits of the data element effectively form a circle and hence the instruction is called a rotate. This is demonstrated in Figure 1.

The shift instruction is slightly different. In fact, there are two types of shift instructions. The first, the Logical Shift, involves always moving a zero into the final bit as is shown in Figure 2. Arithmetic Shift, the other type of shift instruction, pulls a zero into the final bit when a left shift is performed and is hence identical to Logical Shift Left. However, when an Arithmetic Shift Right is performed the instruction does not change the value of the final bit but leaves it containing whatever value it had before the shift was performed (see Figure 3).
The syntax of these instructions is presented below.

ROL: Roll Left

This instruction has two forms depending upon whether the data element being rotated is a memory location or a data register. If the data is in memory location then the instruction has only one operand.

Syntax:
rol {(An)(An)+-(An)x(An)x(An,xr.s)x}

The data size in this case is always *WORD* and the rotate is one bit.

If the data element to be rotated is in a data register then the instruction has two operands. The second is the data register to rotate while the first operand specifies how many consecutive rotate instructions to perform.

Syntax: rol { xDn}, Dn

If the immediate mode is used then x may be any value between 0 and 7. In this case the data size may be *BYTE*, *WORD* or *LONGWORD*.

CCR status:
X Not affected
N Set if most significant bit of the result is set or otherwise cleared
Z Set if result is zero, otherwise cleared
C Last bit shifted out of the operand
V Always cleared.

ROR: Rotate Right

This instruction is exactly analogous to the ROL instruction except that the bits are shifted in the operate direction. The syntax and CCR effects are the same.

The four shift instructions all share a common syntax with the rotate instructions and so it will not be repeated. These instructions do, however, vary in how they affect the CCR as shown below.

ASL: Arithmetic Shift Left

CCR status:
X Set according to the last bit shifted out
N Set according to most significant bit of result
Z Set if result is zero, else cleared
C Last bit shifted out of the operand
V Set if the high bit is changed at any time during the shift, else cleared

ASR: Arithmetic Shift Right

CCR status:
X Set according to the last bit shifted out
N Set according to most significant bit of result
Z Set if result is zero, else cleared
C Last bit shifted out of the operand
V Always cleared

LSL: Logical Shift Left

CCR status:
X Set according to the last bit shifted out
N Set according to most significant bit of result
Z Set if result is zero, else cleared
C Last bit shifted out of the operand
V Always cleared

LSR: Logical Shift Right

CCR status:
X Set according to the last bit shifted out
N Set according to most significant bit of result
Z Set if result is zero, else cleared
C Last bit shifted out of the operand
V Always cleared

The Example Program

Given the above you should now be in a position to understand how our second example program functions. To help you along, the source code has been liberally sprinkled with comments to highlight the main features of the code.

When you've played with the program and satisfied yourself that you know how it works, then why not try modifying it. You could, for example, add an extra proportional gadget to allow the selection of any of the screen's colours can be selected and modified rather than just the background colour. In fact, you could produce a useful utility which could be called up at any time and allow the screen colours to be set without loading preferences. *YA*

GURU'S GUIDE

(Continued from page 47)

memory expansion on a different priority of the 68000's queue, the possible outcome of this will be the keyboard locking up, the mouse going over a loop (or just flickering in one place) or, if I can get in control of the situation, issue a Guru message and try to save the day with a guru message.

All memory expansions are compatible with this neat little package, I strongly recommend you get a copy of this disc as soon as you or anyone out there in Amiga land gets their memory upgrade as, I am afraid that this sort of problem is likely to occur more often.

The package is available from George Thompson Services, Freepost, Dippen, Brodick, Arran, Scotland KA27 8BR for £9.99

Disc Dilemma

Dear Guru

I know you deal with the more technical aspects of the Amiga, but could you please tell me why I have less memory when using my second drive, and why Star Wars fails to load sometimes when I have this drive fitted?

I have an Amiga 500 — is this a problem specific to my own Amiga system?

Mr P Lavers, Puckeridge, Herts.

No problem Mr Lavers, but what is it about you Amiga owners in Herts? Do all of you own Star Wars?

But seriously, running a second or a third or even a fourth drive will take up some memory because the Amiga's operating system — AmigaDOS — is intelligent, and because it is intelligent, it needs to have some memory set aside for its own applications and buffering and so on...

The reason why some applications crash because you have additional drives fitted to the system is because some programs use the additional disc buffering space for their own space, this will result in a pretty amazing looking crash as the screen fills up with garbage and it flashes. Since a lot of information is screen data with some games, it looks like the game is being quickly scrolled through.

The answer? Unplug your drives!

Some disk drive manufacturers have realised that drives may frequently require unplugging to return memory to the user. They have got around this by fitting a switch on the disk drive that totally disables it, freeing up all of that lost memory — without you having to grovel around the back of your machine to physically unplug the drive. One manufacturer that has a drive on the market at the moment with a switch is Cumana, their drives are widely available from most Amiga dealers.

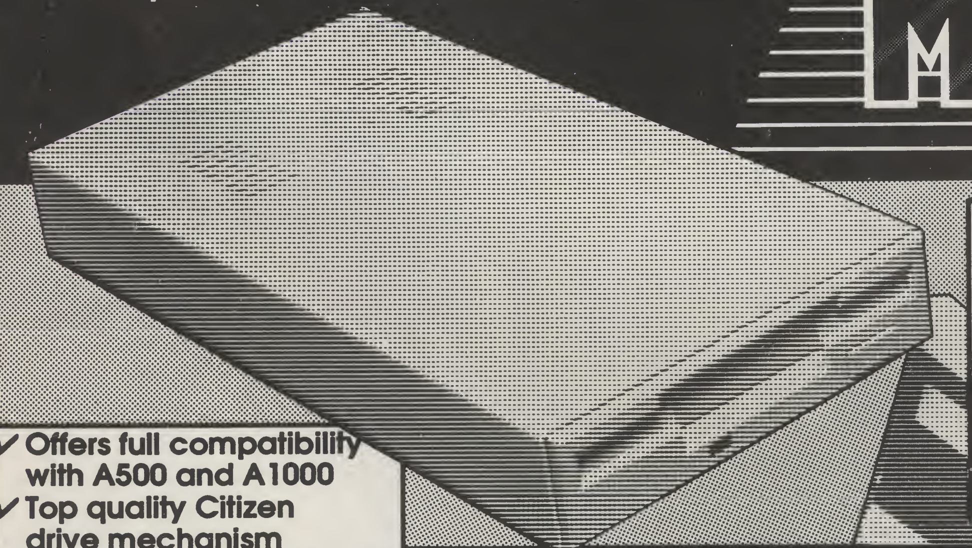
Should you already own a second drive, as you do Mr Lavers, Trilogic offer a switched lead that fits between your Amiga and your 2nd disk drive.A switch on the lead is used to turn the drive off. The lead costs just £9.99 and is available from **Trilogic** at, Dept Y.A, Unit 1, 253 New Works Road, Bradford, BD12 0QP, Telephone: 0274 691115. If you do own a second drive then I think that this lead is worth its weight in gold, it will not only save wear and tear on the drive lead and sockets but also save you from *accidentally* plugging or unplugging your drive while the machine is on and possibly blowing the computer.

There is a Poke I can offer you, but I am exploring the possible ramifications, of this, watch this space.

Visiting Hours:
The Guru cannot answer all your questions in person, so if you have a technical query, why not drop him a line at:
Guru's Guide, Your Amiga, 1 Golden Square, London W1R 3AB.
Questions regarding software piracy will be sent onto the relevant software houses.

High Quality 2nd Drive for the Amiga
- at a price that makes sense!

Evesham Micros



NEW LOWER PRICE !
£89.95
including VAT and delivery

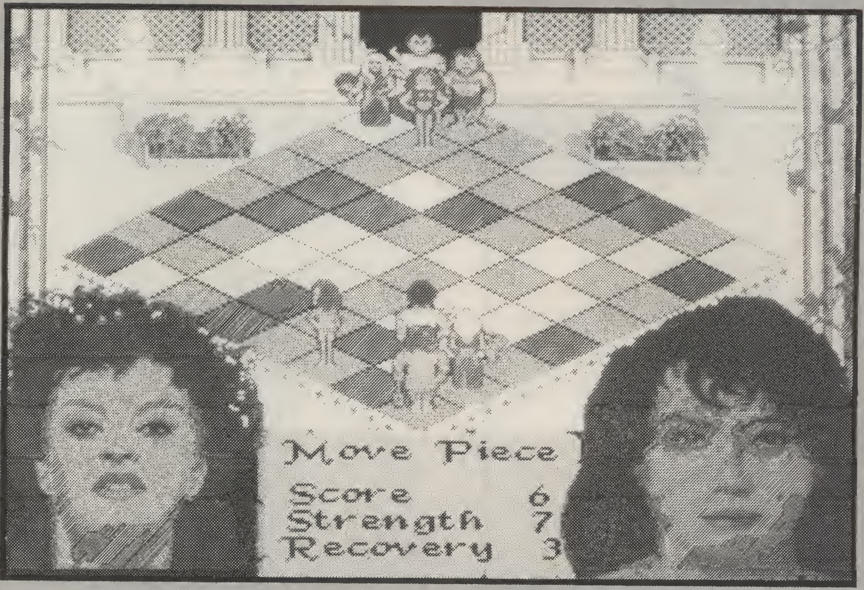
- ✓ Offers full compatibility with A500 and A1000
- ✓ Top quality Citizen drive mechanism
- ✓ 880K formatted capacity
- ✓ Very quiet
- ✓ Slimline design
- ✓ Throughport
- ✓ Long cable for location either side of computer
- ✓ Full 12 months guarantee

All prices VAT/delivery inclusive. Next day delivery £5.00 extra.

How to order...
Send cheque P.O. or ACCESS/VISA details
Phone with ACCESS/VISA details
Govt., educ. & PLC official orders welcome
Same day despatch whenever possible
Callers welcome, open 6 days 9.30-5.30
All offers subject to availability, E.&O.E.
TELEX: 333294 FAX: 0386 765354

Evesham Micros Ltd
63 BRIDGE STREET
EVESHAM
WORCS. WR11 4SF
Tel: (0386) 765500

ALSO AT: 1762 PERSHORE RD., COTTERIDGE, BIRMINGHAM, B30 3BH. TEL: 021 458 4564



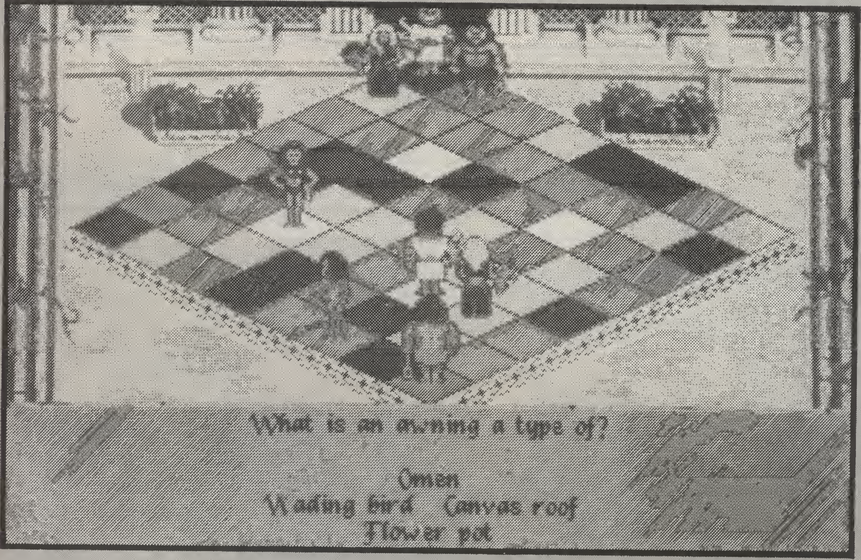
Move Piece
Score 6
Strength 7
Recovery 3

Powerplay

Fin Fahey takes on the gods themselves. Who says they're omniscient?

■ **When it comes to the question of playing** *Trivial Pursuit*, I'm a complete hypocrite. I always claim loudly how much I consider it a small-minded game for empty-headed yuppies, exhibits a stamp collector's approach to knowledge etc etc. As soon, however, as someone digs out a set, I'm down there with the worst of them, trying to dodge the Sports and Entertainment boxes while capitalising on the Science.Imagine my delight, then, when *Powerplay* drops on to my desk. Not only all the fun of TP without those annoying dorks who know all the cards off by heart, but with this one you get to take on the gods themselves...

Powerplay combines the dubious thrill of a general knowledge quiz with boardgaming. Instead of the TP-type racetrack with radial channels, this board is far more chess-like. The players — unfortunately only two can play — each control four animated figures of demigods which they can move across the board when a question is answered satisfactorily. The board squares are in the familiar four colours and correspond to questions about Science, Geography & History, Sports & Leisure and General Knowledge. In addition, there are four teleport squares — if you land on these, you can end up on another random teleport square.



What is an awning a type of?
Omen
Wading bird Canvas roof
Flower pot

Should you try to move on to a square occupied by your opponent, the crucial part of the game takes place — the two pieces square off for a challenge, where each piece takes place in a race against time to answer a question before the other. The piece that fails either gets dropped into a lava pit, or turned into stone by the Medusa. This may not be lethal, since pieces, once they've accumulated enough 'wisdom points' by anwering questions, go up a level and become a more powerful demigod. Failure in a challenge only drops the piece down one level, but if it's on the lower rung then it's dead.

Time is of the essence. Not only do you have to drop in an answer before your opponent, whether human or computer, but the number of wisdom points gained drops if you take too long answering. Pieces also have strength points depending on their level. These come in useful in challenges. Since the winner of a challenge is the player who first answers three questions in a row before the opponent, it is possible to get a stand-off. The strength points decide these. A piece drops one point for each question, so hitting zero means a sudden quietus.

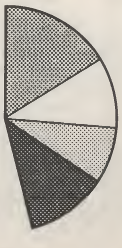
The pieces do recover their strength points in time, depending on their names. Either player can be one of four gods: Hecate, Apollo, Hermes and Aphrodite — Arcana has used the Greek names throughout — and there is a 'family' of four pieces associated with each god.

What of the questions then, since these are the root of the whole game? I can't be sure how big the question database is, but I do feel that a computer implementation of a quiz should improve on the original game in this respect. *Powerplay*, however, failed to impress me — it took little time before I was getting the same questions, or variations on them, thrown back at me. Worse, some of them are very badly thought out, having no proper answer or being ambiguous. An example, which I got several times incidentally, is 'What is the biggest island in the West Indies?'. Answers supplied were: Jamaica; Cuba; Bermuda and the Dominican Republic. The program favoured Cuba. Now leaving aside the fact that the Dominican Republic is not an island, this is at best bad wording. Cuba is the biggest of the islands in the question, but not in the West Indies. That honour falls to Hispaniola, composed of the Dominican Republic and Haiti. The database abounds with examples like these, which completely ruin the fun of playing.

Powerplay is ultimately a let-down. The gameplay is original, if a bit simple-minded, and I like the animations. It falls down on the most crucial part, though — the questions. It took two measly games for me to lose interest. Enough said? *YA*

POWERPLAY

Title: **Powerplay**
Supplier: **Arcana**
Price: **£**



▩ Graphics: **16**
☐ Sonics: **10**
▨ Gameplay: **8**
■ Value: **12**

MAIL ORDER
For all your
software
needs

Cottage Software

All prices
include VAT
and
delivery

GAMES/STRATEGY

Aargh	£15.95
Annals of Rome	£19.50
Art of Chess	£18.95
Backlash	£13.50
Bad Cat	£20.79
Balance of Power	£22.97
Battleships	£14.44
Brattaccus & Arena	£21.27
Casino Roulette	£14.49
Dejavu	£20.95
Ebonstar	£19.50
E.C.O.	£17.95
Flight Simulator 2	£34.19
Flintstones	£14.44
Football Manager 2	£15.95
Galactic Invasion	£17.95
Garrison	£20.85
Hardball	£21.88
Indoor Sports	£18.82
Jet	£28.95
Lancelot	£14.25
Leaderboard	£18.43
Mercenary Compendium	£18.06
Pink Panther	£16.45
Police Quest	£15.95
Rolling Thunder	£19.75
Silent Service	£17.95
Space Quest 2	£15.95
Stellar Conflicts	£21.95
Tetra Quest	£14.25
Three Stooges	£23.95
Time & Magik	£14.17
Uninvited	£21.68
Vampires Empire	£15.94
Winter Olympiad 88	£14.16
Zenon	£18.95

BUSINESS/UTILITIES

Amiga A500 Package	£385.00
Amiga Basic Inside & Out	£14.99
Amiga Machine Language	£11.50
Amiga Tricks & Tips	£11.50
AmigaDOS Manual (2nd edition)	£19.75
Baxall Camera	£199.00
Cherry A3 Digitizing Tablet	£540.00
Cherry Tablet Driver	£29.00
Digipic Frame Grabber	£249.00
Easyl Drawing Tablet	£279.00
Hitachi 650 Line Camera	£259.00
NEC External 3.5" Drive	£92.95
Professional Genlock	£745.00
PAL Genlock	£259.00
Handy Scanner	£265.00
Supra 20mb External Drive	£599.00
Supra 60mb External Drive	£1240.00
Canon 1080A Color Ink Jet	£499.00
Epson FX800 Matrix	£439.00
Epson LX800 Matrix	£285.00
Juki 5510P Matrix	£285.00
Juki 5520P Matrix	£460.00
MP 135 Matrix	£159.00
MP 135+ Matrix	£169.00
MP 165+ Matrix	£209.00
MP 200+ Matrix	£310.00
NEC P6+ (24 Pin)	£650.00
NEC P2200 (24 Pin)	£390.00
Xerox 4020 Color Ink Jet	£1150.00
10 x 3.5" Disks (DD/DS)	£11.50
Disk Box for 40 Disks (+lock)	£8.50
Disk Box for 100 Disks ("")	£18.50
Mouse Bracket	£3.50
Mouse Mat	£4.50
3.5" Disk Drive Cleaner	£5.99
Amiga A500 Dust Cover	£7.50

Aegis Animator Images	£79.50
Aegis Audiomaster Sampler Ed.	£33.65
Aegis Images	£21.46
Aegis Sonix	£38.95
Aegis Videoscape 3-D	£103.95
Aegis Videotitler	£77.25
Amiga Prosound Designer	£59.95
Amiga Ruby View/Term	£73.95
Amiga Synthia	£69.95
Award Maker	£28.95
Aztec C Compiler (Comm.)	£372.28
Aztec C Compiler (Prof.)	£141.95
Beckertext	£87.73
Businessware 1 & 2 (each)	£85.00
Butcher	£20.95
Calligrapher Colorfonts	£59.95
City Desk Desktop Publishing	£79.95
Deluxe Music Contruction	£51.36
Deluxe Paint 2	£50.95
Deluxe Video (PAL)	£51.20
Descartes	£22.37
Digidroid	£52.48
Digipaint	£42.95
Digiview 3	£124.95
DOS to DOS	£28.95
Expert Systems Kit	£51.15
Express Paint	£47.50
Forms in Flight	£52.95
Futuresound	£158.85
Galileo	£42.95
Hisoft Devpak	£44.95
Intro-CAD	£47.95
K-Seka 68000 Assembler	£36.95
K-Spread 2	£56.86
Karafonts	£42.95
Lattice AmigaDOS C V4	£131.65

HARDWARE

Lions Fonts	£28.95
Logistix	£84.11
Marauder 2	£28.74
Mathamation	£51.22
Maxiplan A500	£74.95
Maxiplan Plus	£109.73
MCC Macro Assembler	£52.95
MCC Shell	£37.95
Modula 2 Standard System	£74.50
Page Flipper	£26.95
Page Flipper plus F/X	£87.95
Pagesetter (Gold)	£69.95
Perfect Sound	£69.50
Photon Paint	£52.95
Power Windows 2	£51.95
Prism Plus V1.2 (PAL)	£42.95
Pro Video	£116.90
Pro Video Fonts Library	£50.26
Pro Write 2	£64.95
Publisher Plus	£72.50
Quizmaster	£56.95
Scribble 2	£39.95
Sculpt 3-D	£55.95
Sculpt 3-D Animate	£93.38
Silver Ray Tracing	£87.73
Softwood Write & File	£51.95
Soundscape Midi Interface	£43.95
Soundscape Promidi Studio	£125.95
Soundscape Sound Digitizer	£92.95
Studio Magic	£51.95
Superbase Personal	£74.95
True Basic 3-D Graphics	£29.95
True Basic Arithmetic	£29.95
Turbo Silver 3-D	£104.95
TV Show & TV Text (each)	£69.95
Zumafonts	£25.95

★★★★★★ SPECIAL OFFERS ★★★★★★

THE POWER BUSINESS
Vizawrite
Logistix
Superbase Personnel
PACKAGE PRICE £174.95

COMPLETE DESKTOP PUBLISHING PROGRAMME
Vizawrite
City Desk
PACKAGE PRICE £132.95

PROFESSIONAL IMAGE PROCESSING
Digipic
Pixmate
Baxal Video Camera
PACKAGE PRICE £464.95

SOUND SAMPLING PACKAGE
Perfect Sound
ECE Midi Interface
PACKAGE PRICE £115.95

INTRODUCTORY GRAPHICS PACK
Express Paint Page Flipper
Introcad
PACKAGE PRICE £86.95

MORE MEMORY — MORE MANAGEMENT PACKAGE
Supra 20mb Hard Drive
Diskmaster
PACKAGE PRICE £597.95

Just because you don't see it — doesn't mean we haven't got it
For full details, send a large stamped addressed envelope
Please make cheques and postal orders payable to:
"COTTAGE SOFTWARE"
Please endorse cheques with bankers card number
P.O. Box No. 8, Shaw, OL2 8QN Telephone: 8706 845365

MPC SOFTWARE

Leisure Software

Carrier Command	£ 19
Chessmaster 2000	£ 19
Defender of the Crown	£ 19
Empire	£ 20
European Scenery Disc	£ 19
Ferrari Formula 1	£ 20
Flight Simulator II	£ 32
Flight Simulator II scenery disc 7 or 11	£ 19
FS2 Japan Scenery Disc	£ 19
Football Manager 2	£ 13
Grid Iron	£ 19
Hollywood Strip Poker	£ 8
Interceptor	£ 24
Jet	£ 30
King of Chicago	£ 21
Pandora	£ 19
Quizzam	£ 18
Return to Atlantis	£ 17
Sentinel	£ 16
Star Fleet 1	£ 18

Arcade Games

Barbarian (psygnosis)	£ 15
Empire Strikes Back	£ 17
Garrison (1 or 2)	£ 19
Insanity Fight	£ 10
Marble Madness	£ 17
Motorbike Madness	£ 12
Offshore Warrior	£ 20
Peter Beardsley Football	£ 16
Pinball Wizard	£ 17
Return to Genesis	£ 18
Skyfox	£ 10
Starglider 2	£ 19
Star Ray	£ 18
Stir Crazy	£ 16
Winter Games	£ 10
Vixen	£ 15
Wizball	£ 19
Xenon	£ 17

Business Software

Acquisition	£ 200
Excellence	£ 190
Intellitype	£ 20
Intro CAD	£ 50
Kind Words	£ 45
K-Spread 2	£ 55
Maxiplan 500	£ 80
Maxiplan +	£ 120
Professional Page	£ 220
Pro Write	£ 60
Pro Write 2	£ 85
Ruby-Comm	£ 100
Superbase Personal	£ 85
Superbase Professional	£ 210
The Works	£ 95
XCAD	£ 400

Books

Advanced Amiga Basic	£ 17
Amiga Applications	£ 17
Amiga Basic Inside & Out	£ 17
Amiga for Beginners	£ 17
AmigaDos Manual V2	£ 23
Amiga Machine Language	£ 15
Amiga Machine Lang Guide.	£ 20
Amiga Programmers Guide	£ 17
Amiga Programmers Handbook Vol 2	£ 24
Becoming an Amiga Artist	£ 19
Big Red Book of C	£ 8
C Programming Language (K&R)	£ 23
Hardware Reference Manual	£ 23
Inside Amiga Graphics	£ 17
Inside The Amiga with C	£ 22
Intuition Reference Manual	£ 23
Kickstart Guide	£ 11
Kids & The Amiga	£ 17
New Hackers Handbook	£ 8
Programmers Guide to the Amiga	£ 24
ROM Kernel: Exec	£ 23
ROM Kernel: Libraries & Devices	£ 33

Please phone for post rates for outside the UK

Adventures

Bards Tale 1 or 2	£ 21
Corruption	£ 19
Jinxter	£ 19
Knight Orc	£ 8
Legend of the Sword	£ 18
Romantic Encounters at the dome	£ 19
Shadowgate	£ 19
Time & Magic	£ 16

New Products

BAT	£ 20
Battle Chess	£ 20
Captain Blood	£ 19
Carrier Command	£ 19
Dungeon Master	£ 19
Elite	£ 19
Fusion	£ 20
Iron Lord	£ 20
Lancelot	£ 17
Menace	£ 19
Netherworld	£ 19
Publishing Partner Professional	Phone
Rocket Ranger	£ 24
Sargon 3 Chess	£ 19
Skate or Die	Phone
Virus (zarch)	£ 16
Whirligig	£ 18
Zynapse	£ 16

Programming Languages & Utilities

Aztec C Developer Version	£ 250
CLimate	£ 35
DevPac	£ 48
K-SEKA	£ 48
Lattice C V4.0	£ 150
Metacomco Macro Assembler	£ 55
Metacomco Pascal	£ 70
Project D	£ 35
Zing	£ 60

Sound & Graphics

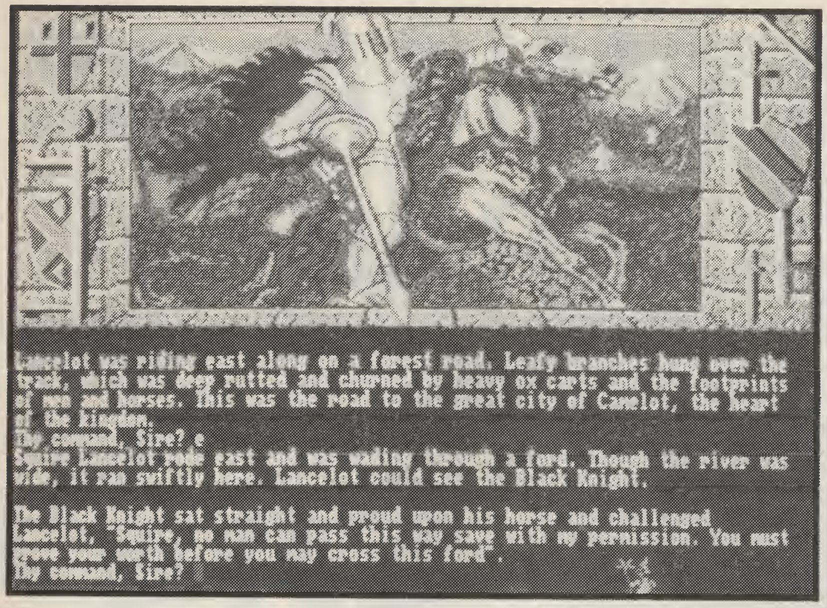
Aegis Audio Master	£ 40
Aegis Sonix	£ 50
Aegis Video Titler	£ 85
Animate Apprentice	£ 140
Animate Junior	£ 50
Animate 3D	£ 100
Butcher 2 PAL Version	£ 23
Deluxe Paint II PAL	£ 60
Deluxe Photolab	£ 60
Deluxe Print	£ 18
Deluxe Print 2	£ 40
Deluxe Productions	£ 95
Deluxe Music PAL	£ 50
Deluxe Video 1.2 PAL	£ 60
Digipaint	£ 55
Director	£ 40
Fantavision	£ 39
Forms in Flight	£ 65
Grabbit	£ 25
Instant Music	£ 18
Lights, Camera, Action!	£ 50
Newsletter Fonts	£ 25
Photon Paint	£ 65
Photon Paint Expansion Disc	£ 18
Pixmate	£ 40
Sculpt 3D	£ 75
Soundscape Utilities #1	£ 30
Studio Fonts	£ 25
TX*Show	£ 60
TV*Text	£ 60
Videoscape 3D	£ 100

Deluxe Data Discs

Arts Parts #1
Arts Parts #2
Hot & Cool Jazz
It's only Rock 'n' Roll
Seasons & Holidays

£ 8 each or any 3 for £ 21

MPC Software,
72, Julian Rd., West Bridgford, Nottingham NG2 5AN
Tel: (0602) 455114
Mail Order Only



```
Lancelot was riding east along on a forest road. Leafy branches hung over the
track, which was deep rutted and churned by heavy ox carts and the footprints
of men and horses. This was the road to the great city of Camelot, the heart
of the kingdom.
Thy command, Sire? e
Squire Lancelot rode east and was wading through a ford. Though the river was
wide, it ran swiftly here. Lancelot could see the Black Knight.

The Black Knight sat straight and proud upon his horse and challenged
Lancelot, "Squire, no man can pass this way save with my permission. You must
prove your worth before you may cross this ford".
Thy command, Sire?
```

LANCELOT

In days of old when knights were bold... Gordon Hamlett tries to be both chivalrous and brave simultaneously.

■ **Considering how little is actually known** about him, it is amazing to see the amount of literature written about King Arthur. The exploits of his Knights of the Round Table have received the full treatment from Hollywood, classic authors (Mark Twain), historical novelists, fantasy novelists and historians. As well as films and books, Wagner wrote a series of operas based on the legends and now, moving into the hi-tech age, we have the computer game.

Level 9 in conjunction with Mandarin Software have released *Lancelot*. This three-part adventure is based on the tales of King Arthur's greatest knight, although a little artistic licence has been thrown in by using the stories more usually associated with Sirs Gawain and Tristram but this is no great thing when compared to some of the liberties taken by other authors.

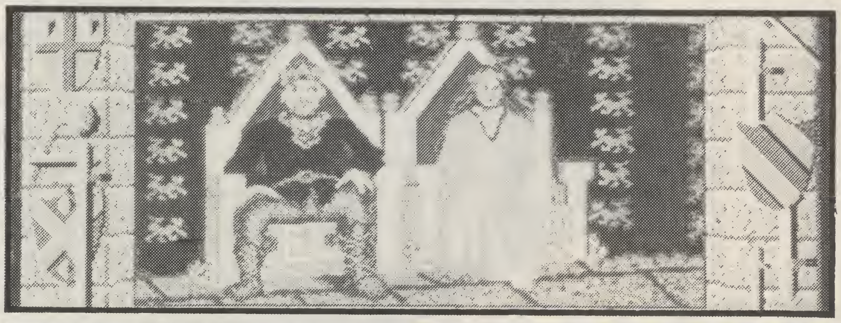
Lancelot is one of the great tragic figures of literature. He soon became renowned as the greatest knight in the land, his feats of courage second to none. He was less successful with women though. An ill-starred affair with the fair Elaine begat Galahad, the purest knight of them all and the only one to eventually achieve the quest for the Holy Grail. His second great love though was even more unfortunate in so much as it was with Guenever, Arthur's Queen. As the affair became open knowledge, so the Arthur's enemies used it to drive a wedge between the king and his favourite knight. This in turn led to rifts that eventually split the company of the Round Table asunder and Lancelot, rather than riding off into the sunset with his best damosel by his side, got the abbey habit at Glastonbury and became a bishop.

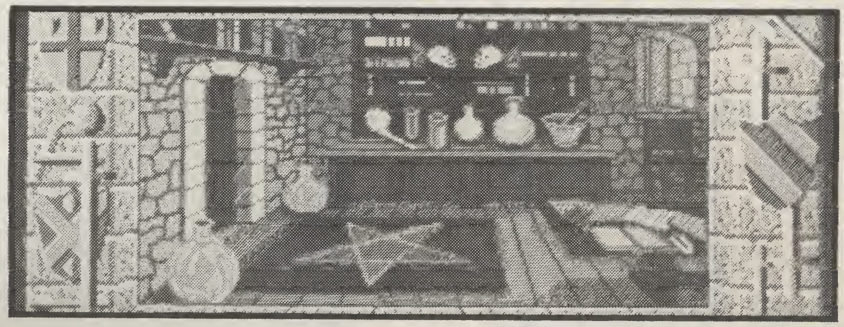
The game centres round three main stories. Lancelot must first make his way to Camelot and be knighted by Arthur. Then come a series of mini-quests to prove that you really are the greatest knight. As well as fighting evil, you must rescue the knights that have managed to get themselves captured and return them to the Round Table. In part two, you need to ensure that your son Galahad can come to Camelot to take his place in the Siege Perilous — he is the only knight worthy of sitting there. As the Round Table is now complete, so the quest for the Holy Grail can now take place. Because of his previous sins, Lancelot is only granted a vision but Galahad can progress further.

The text is vaguely mediaeval, in keeping with the style of Malory who wrote *Morte D'Arthur* back in 1485. This was not the first mention of Arthur's existence but the book upon which most of the other stories are based. Nevertheless, a few examples of Level 9's humour do occasionally creep in, especially with the town crier. Announcements that war has broken out in France but no-one from Camelot has been hurt smacks verily of Ye Olde News at Tenne.

The descriptions are on the whole good, although the story suffers somewhat from an absence of scale — a problem common to most adventures. Thus while there are some thirty locations within Camelot, it is only three steps outside the town (located in Southern England) before you reach Scotland! Perhaps if the text had been prefaced with 'After a week's hard riding, you arrive at ...' would have helped.

Throughout the text, certain words appear CAPITALISED. This is because they represent clues to the whereabouts of a genuine jewel encrusted goblet worth some £5000 and no mean prize for anyone with a mind devious enough to unravel the solution.



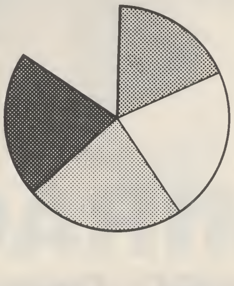


The graphics are of a reasonable standard although there is a fair amount of repetition. Personally, I prefer the style adopted by Magnetic Scrolls in which only a few locations are illustrated but in great detail.

Minor quibbles apart though, I enjoyed *Lancelot* more than most other adventures that I have seen recently and anyone who has ever imagined themself going into a joust with a lady's favour tied round his lance is surely going to want this for their collection. *YA*

Lancelot

Title: **Lancelot**
Authors: **Level 9**
Suppliers: **Mandarin Software**
Europa House
Adlington Park
Adlington
Macclesfield
SK10 4NP
Tel: 0625 879940
Price: **£19.95**



▨ Graphics: 18
☐ Story : 22
▨ Gameplay: 23
■ Value : 21

Access

2ND AMIGA DISK DRIVE

VISA

NEC mechanism
880K capacity

By selling direct we can provide the best price and maintain the highest quality. Rest assured that if, for any reason, you do not wish to keep the item, then return it to us within 14 days of purchase and we will refund your money in full. Should any item purchased from us fail during the first 12 months then we will repair it free of charge.

2ND PRINTER INTERFACE

Use your Amiga's serial port to drive a second Centronics compatible printer. A 3 metre cable is included - no extras required.

MIRACLE SYSTEMS

NBC, Dean Road, Yate, Bristol BS17 5NH
Telephone orders welcome on (0454) 317772

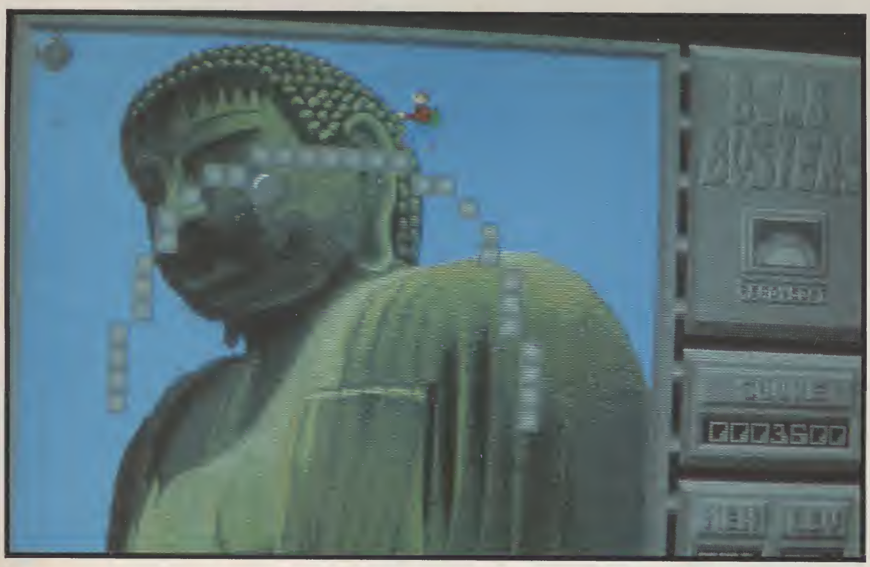
Bomb Busters

Remember Bomb Jack the coin op? — Karen Young takes a look at a lookalike

Bomb Busters is an unwelcome return to the old "coin op classic" Bomb Jack, the archetypal ladders and ledges game, whereby you had to jump around, defusing and disposing bombs, keep an eye on any bombs that hadn't gone off yet, and generally running about a lot as one tends to find one doing in the middle of an arcade scenario.

I will say now that *Bomb Busters* is remarkably slow for the amount of computer power actually needed to produce a game of this standard, each screen takes about twenty seconds to load in, and boy, what a wait! You would have thought some companies had enough sense to use the Amiga's 512k memory map and load a few screens from memory, but no, *Bomb Busters* takes an age to play — a time totally disproportionate to the amount of time spent playing each screen! The backgrounds are nice — being very pretty illustrations of places around the world, like the Taj Mahal, the Pyramids and the Sphinx, and the screen clearing routines are pleasant to peruse, but the character definition and movement is pretty awful.

Sorry Readysoft, but what you have here is a very insubstantial game. It gets repetitive after a while — which is, I suppose, acceptable, as there are other features to make the game worth playing. *Bomb Busters* plays like it is written in Basic, and it feels as if the game is hurried. Not even the really wonderful slabs of heavy metal music slapped across the loading of the game look as if they can save the disk being formatted and used elsewhere.



It takes all of 15 seconds to complete screen one, but it takes 20 seconds (approx) to load in another screen! It is about time somebody wrote a decent speed package that worked with games like this, whose protection wipes out any "go-faster" software, because even though the game is a bit cruddy (to put it mildly), things could have been greatly improved by a faster disk drive operation and a better set of animation routines.

The character — unnamed — but he was called "Jack" in the original coin-op, so for legal reasons we will call him "John" (Bomb John... now there's a suggestive name if ever I heard one!) is on a flying pack, you know, the ones used in *Lost In Space* for the first Black and White season. He flies about (to really realistic sound effects, honestly!) defusing bombs and walking into extra bonus points — if you are good, you will get a bonus score at the end of a section, but I literally lost patience with this game after a day or two.

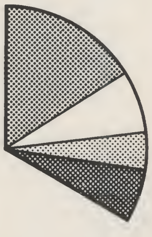
Even the addition of a second drive will not help matters much as the game doesn't access the drives all of the time, it only loads in data as, and when, necessary.

So, apart from looking at a rigid little man on a backpack making lots of flatulent noises, then I would give this a miss. I reckon it is time for the original coin-op version to make a proper conversion over to the 16-bitters as the original had a lot going for it: full colour; lots of sound; enjoyable muzak; plenty of things to keep the eyes and the wrists on the move.

This version of *Bomb Busters*, however, is a poor imitation. It is very slow (although the gameplay is moderately fast — it's just the loading of different screens). Come on ReadySoft! ***YA***

BOMB BUSTERS

Title: **Bomb Busters**
Supplier: **ReadySoft**
25 Red Oak Drive
Richmond Hill
Ontario
L4C 4X9
Canada
Tel: 416-731 4175



Graphics: **16**
Sonics: **7**
Gameplay: **4**
Value: **6**

The Empire Strikes Back

For an exorbitant transfer fee the A&B Brothers Journey to Hoth for a bit of wireframe blasting

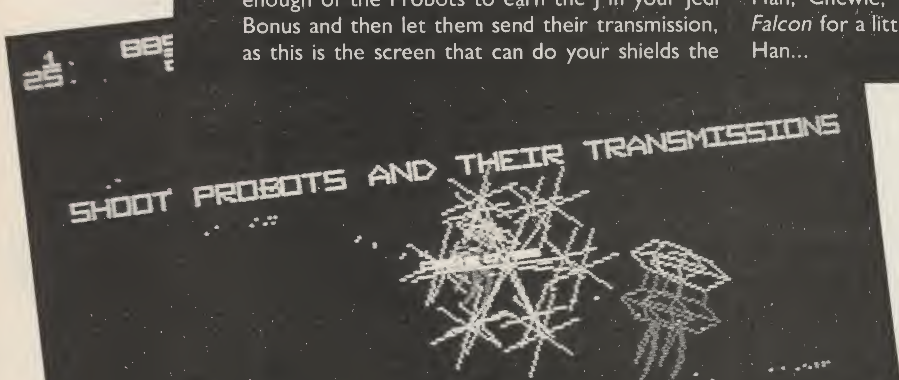
■ **Many people weren't aware that *The Empire Strikes Back*** was ever an arcade game. To fill in the the time between the launch of *Star Wars* and *Return of the Jedi*, Atari's coin-op division kept the arcade owners quiet by offering a new set of ROMs to go in existing *Star Wars* machines, which generated the new scenario. However Domark has decided not to sell *Empire...* in the form of a data disk for *Star Wars* but as a totally separate package. A policy that many have questioned. Personally, I think that marketing the game separately is the only thing that could be done. So is there enough meat on this game to keep a young Jedi happy (apologies at this stage if there are any vegetarian Jedis out there)? Let's leap into my snowspeeder and shoot off on a routine patrol.

Oh, No! I've spotted a large number of Imperial Probots sent down to suss out where our secret base is. It's up to me to blow them away before they send all four parts of their transmission to the Star Destroyers in orbit around Hoth.

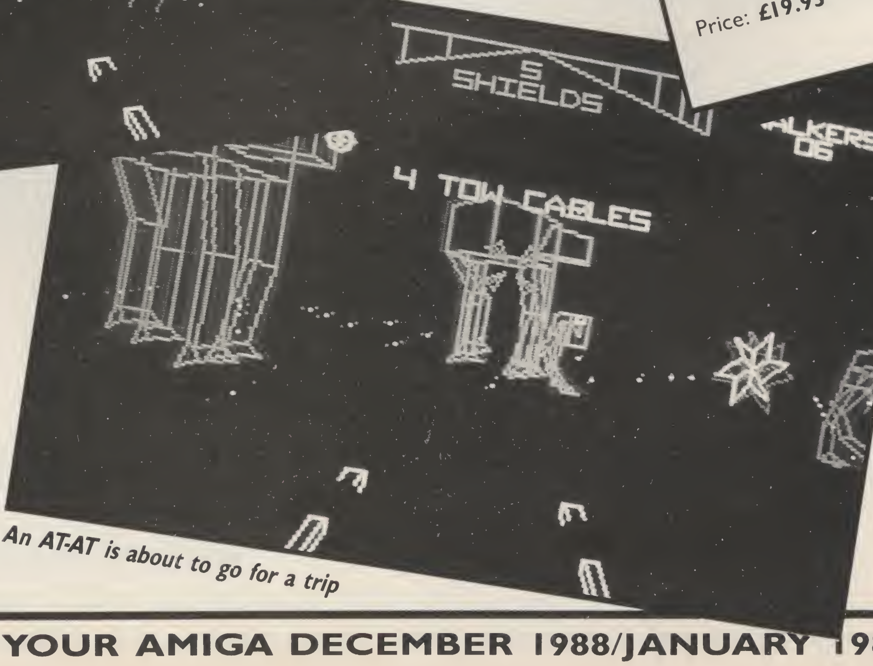
The best tactic seems to be to shoot just enough of the Probots to earn the J in your Jedi Bonus and then let them send their transmission, as this is the screen that can do your shields the most damage (and you've only got five at your disposal).

Once the tranmission has been sent along come the two-legged AT-ATs and the dinosaur-like AT-STs. They slowly advance towards the rebel base, squashing outposts along the way. Although it's impossible to make them turn tail and run you can slow them down by taking out the leaders of the pack. This can be done in two ways. You can either blast both sorts of transport in the red viewing port or, you can use a tow cable to trip up the AT-STs. Bonus points can be scored for trick flying through the walkers' legs. 5000 points are awarded to the first successful attempt, then 10 000 for the next 15 000 for the one after, and so on. One nice trick here is to fly through the legs and release a tow cable a the last minute, so that you get both sets of points. What I can't understand is, how do dirty great AT-STs turn from awesome green giants into three small white boxes when blasted?

The walkers have managed to break down all the outposts and raided the base so it's time join Han, Chewie, Leia and C-3PO in the *Millenium Falcon* for a little brush with the law. Over to you Han...



Watch those screens — here comes the probots

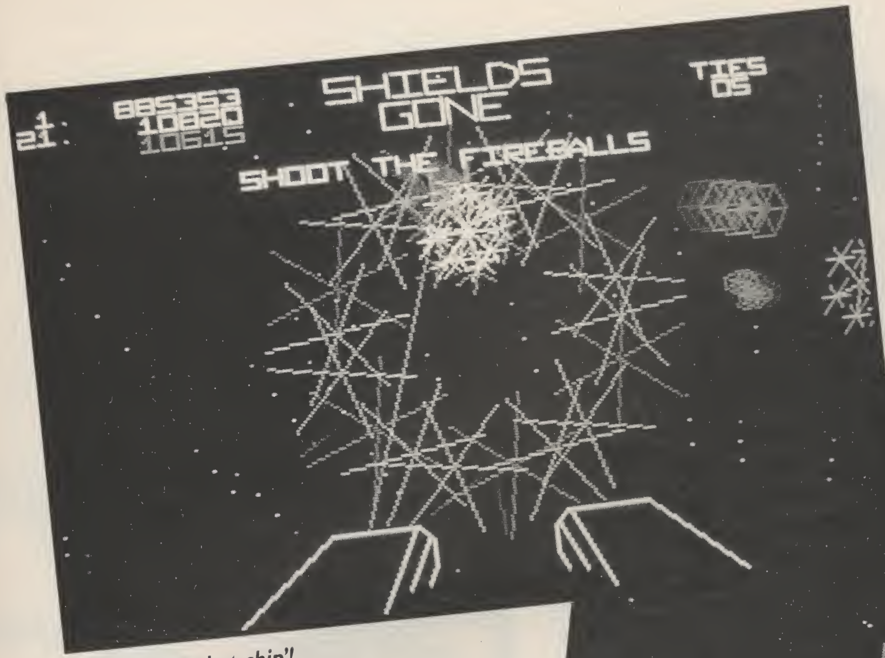


An AT-AT is about to go for a trip

THE EMPIRE STRI

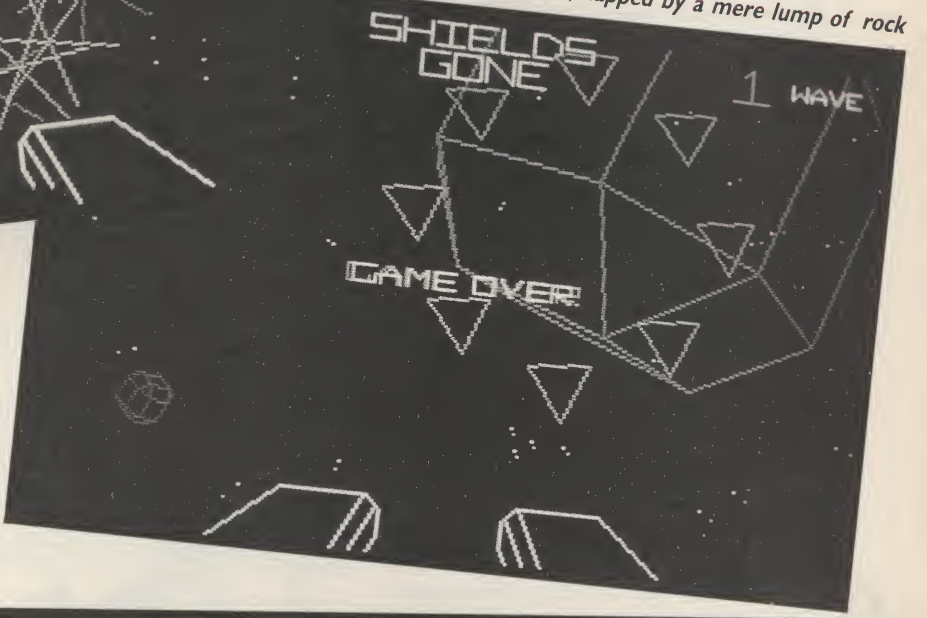
Title: **The Empire Strikes Back**
Supplier: **Domark**
Ferry House
91-57 Lacy Road
London SW15 1PR
Tel: 01-780 2222

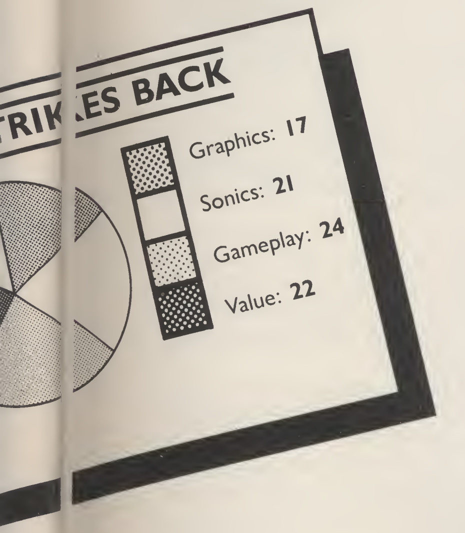
Price: **£19.95**



'I must have that ship'!

...after all that, zapped by a mere lump of rock





TRIKES BACK

- Graphics: **17**
- Sonics: **21**
- Gameplay: **24**
- Value: **22**

Thanks Luke... Playing the game brings me back to the days when we went in on our first attack on the Death Star, the kid and I sure played hell with those TIE fighters, and why not? I doubt if these Imperial guys even have toilets on their star cruisers, or even if they know how to go to the toilet without assistance! After Luke gets past the AT-ATs and the AT-STs he flies over the rebel base and the scene moves over to my side of the planet — it's just like the old days with TIE fighters swerving in front of us all the time lobbing more missiles at us, Chewie and I are pretty good at shooting the missiles, but it doesn't help having that stupid 'droid screaming all of the time. Well, Chewie shuts him up for a while in the hold.

The ships are quite well represented, with some decent sound-effects coming across in stereo, which is great if you have your Amiga hooked up to the hi-fi, The patterns they fly in front of your ship are different to the original *Star Wars* game (didn't seem like a game at the time), and while the general structure of the TIE fighters hasn't changed — you hit them once and they fall apart — their movements have, making the gameplay annoyingly easy to work out. Even at high levels it is just a matter of time until you learn the patterns they form themselves, and all in all, I rather wish a little more care and attention had been taken to this part of the proceedings.

Still, the sound effects are a damn sight better than *Star Wars*, and at least the TIE fighters scream at you as they fly past. I can't understand how — in space, no one can hear you scream. (Wrong film -Ed.)

That was stage three. The last stage was quite a tricky one, and even old Chewie nearly lost his cool. Me? I lost my best pair of pants. Unfortunately, that stupid 'droid managed to get out.

I am still pretty annoyed at the Princess for calling me a "Scruffy Nerf herder" — I am not scruffy, heck, we're doing half a parsec flat out(at sublight — we ought to change that old speed measurement). She stays shut up and all that bucket of bolts can do is say things like "One more hit and we're doomed!" What a farmer's boy was doing with a protocol 'droid in the first place I don't know...

Anyway, heads up to the display and a load of asteroids are hurling at you — naturally we're doing sublight and with Chewie keeping quiet for once I'm the unlucky one with the controls, shields're in a reasonably good condition, but you can't fire at the rocks — they just hit your shields. I find that just flying down in one corner (the bottom left hand corner) saves me from most of the debris. Chances are that you get hit once or twice at most this way. All I do is wait it out until old tinman to my right tells us "Thank goodness, we're coming out of the storm", so he has a use after all! I can't use him to tweak the Falcon though, he spends most of his time teaching the ventilation system Bocci, he says it'll help it with the "dialect problem".

The dialect problem just broke down as well... So anyway, once that's finished there's a rather nifty shot of the Falcon flying off into the corner with the letter "I" hopefully finishing off the word "JEDI", and then it's back to square one with twenty seconds or so of shields and the kid at the controls again shooting at Probots on the planet Hoth, but this time with more Probots, more enemy fire, and more headaches.

No one said fighting the Empire would be easy! Comment: Stop asking me what I think of it, can't you see I'm at level ten? ***YA***

F-BASIC

Does F-BASIC give your Amiga the speed of a C Compiler? Allen Webb investigates

■ **In my view, one of the nice features** of the Amiga is that since very little is held in ROM, it is an open machine which allows effective use of all types of programming languages. This is further aided by the provision of extensive libraries of routines on disk.

As you will be aware, the Amiga has a Basic bundled with it. This is a very good version, which allows the writing of clear and structured programs. It is, though, rather slow and is therefore limited in its value to the programmer. It is therefore not surprising that users soon start looking for more powerful languages. Due to the way that the Amiga libraries were written, the best choice of high level language is C. To somebody who is new to the machine, this means that a new language must be learnt. Similarly, assembler is a daunting alternative. For those of you who don't want to launch forth into new pastures, the alternative is to use a better Basic. F-BASIC offers such an option.

Given the specification of the Basic language and the additions that AmigaBASIC offers, it is a little difficult to envisage what additional features could be provided by F-BASIC. So far as I can see, two main features are offered:

■ The authors have attempted to improve the interface with the routine libraries and the low level functions of the machine. This has been achieved by pinching ideas from other languages such as C and Forth. More on this later.

■ The language is compiled.

Why a compiler?

For those of you who don't know about compiled languages, here is a little explanation. When you run an AmigaBASIC program, each line is scanned by the interpreter as it is executed. The interpretation process involves translating the high-level text via a parser into the necessary calls to routines. This process takes time over and above that taken by the routines and it takes place every time the line is executed. In the case of compiled languages, the high level code is translated in a single step and is converted into an intermediate or pseudo-code. (Users of UCSD Pascal, for example, use p-code). This code is not necessarily 'pure' machine code, indeed the only way to write optimised machine code is via an assembler. The important point is that compilation only takes place once and the time-consuming overhead of the interpreter is lost.

It is possible to buy compilers for AmigaBASIC but I have my doubts as to whether it is really possible to write an optimal compiler for an existing language. F-BASIC has its own syntax and the compiler appears to be custom written.

The package comprises of a substantial A4-size folder, a system disk and a disk containing some 65 examples. Both disks are unprotected and you are encouraged to use backup copies. The feature that becomes quite obvious is that to use F-BASIC will require some work from the user. The syntax has more than a little resemblance to C and many of the Basic commands have been rethought. The improved performance, however, makes the effort worthwhile. The creation of an F-BASIC program involves two steps:

■ You write the code using a text editor. This can be the CLI ED command, EMACS (on your Extras disk), the AmigaBASIC editor (provided you save the file as an ASCII file). You could use a word processor such as Scribble but I suspect that you would have problems if the WP embeds control codes in the text.

■ The source code is compiled by invoking the compiler from disk. Having used QuickBASIC on a PC I found this non-resident system tedious but it has obvious advantages to those with unexpanded A500s who want to write large applications. Since I have a megabyte available, I simply altered my startup-sequence file to ensure that the necessary files were loaded into RAM disk. This gives a simple system to work with.

The final compiled program comprises of two parts: the compiled object code and a runtime library.

To give you an idea of the appearance of F-BASIC programs I have included two listings. Listing 1 shows a simple graphics program and listing 2 shows that the use of libraries is somewhat similar to C.

Under the Influence

From a quick glance at the listings you will have seen the influence of a number of other languages. Rather in the style of Pascal you need to declare variables by type. The accepted types are the usual Integer, Real and Text. Although most users will normally stick to binary, hexadecimal and decimal, integers can be entered in any of the bases from 2 to 36 — rather like FORTH. Integers of 8, 16 and 32 bit lengths are available. Floating point variables operate to 9 significant digits although double precision is not supported. In view of the limited value of double precision, this is not a major failing. String declarations are handled in a slightly odd way.

Strings are assigned a specific length. If you choose to omit this step, a default length of 128 is used. I suspect that this is to allow the efficient assignment of memory during the compilation process. Old favourites such as MID$ and LEFT$ are omitted. A single command with the following syntax is used:

```
SUBSTRING = STRING(X:Y)
```

This extracts a sequence of characters from STRING starting at the Xth and ending at the Yth and puts it into SUBSTRING. The string-handling capabilities are extended by a set of system functions and macros. In summary these are:

FILLCHAR — Fills a specified string with a specified sequence of characters.

LEVEL — Searches a string for the first occurrence of one of a series of substrings.

REPLACE — Performs a search and replace function on a string.

SUBSTRING — Finds the position of a string within another string.

TXTSORT — Sorts the specified portion of a text array.

It is also possible to search any array (numeric or text) for a specified value by use of IN. These commands are further enhanced with a complex system for pattern matching with strings. The system is too involved to discuss here — indeed I haven't fully mastered it myself. It is sufficient to say that using the text handling facilities within F-BASIC it should be possible to develop a most impressive parser for adventures or artificial intelligence uses.

The TXTSORT is mirrored on the numeric side by INTSORT which sorts the specified portion of an integer array. The system offers a rather satisfying method of introducing data statements. Rather than the usual approach the syntax is as in the following example:

```
DATA
(X,22),(Y,1.234),(NAME,"FRED")

DATA (ARRAY,1,2,3,4,5)
```

When the program is compiled, the variables are assigned the indicated values. This means that there is no need for a loop reading the values at the start of the program. On the minus side, it makes life difficult if you need to restore the values to their original settings.

If all this wasn't enough, F-BASIC provides another storage system using RECORD structures. Records are similar to database records in that they act rather like arrays containing differing variable types. Special variables called pointers can be used to hold the address of records. Using this sort of system, it is easy to set up structures such as linked lists.

Overall F-BASIC uses most of the well known Basic functions. It allows the writing of structured code in a manner similar to AmigaBASIC and has all the normal control structures (IF....THEN...ELSE, WHILE... ENDWHILE, REPEAT...UNTIL, ON...GOTO, ON...GOSUB, FOR...NEXT...EXIT). It additionally offers a WHEN (CASE) statement which is a sophisticated IF...THEN...ELSE. Here's an example:

```
WHEN Val IS
   [1]     PRINT "Val = 1"
   [2]     PRINT "Val = 2"
OTHERWISE
           PRINT "Val > 2"
ENDCASES
```

The value of Val is compared to the values in the square brackets and the corresponding line executed.

The usual program and subprogram constructs are used but F-BASIC is more pedantic in that even the main program segment must be declared. This again is very much in the flavour of Pascal. The system for passing parameters between the main and sub programs and the use of global and local variables is very similar to AmigaBASIC. The program block or unit approach of the language however means that the structures are a little more involved. Recursion is also supported in that each time a sub-program calls itself a new set of local variables is created. For the brave programmer this allows the writing of fast and efficient programs.

Graphics and sound

The routines for graphics and sound are similar to AmigaBASIC but perhaps a little more comprehensive. In reality, however, most worthwhile graphics actions are best performed via library functions.

F-BASIC shows links with C and a general empathy with the Amiga operating system in the way that it uses functions. Consider this example using the function for opening a graphics screen:

```
ADDRESS = SCREEN 1 (0,200,5,1,0)
```

This opens a low-resolution non-interlaced screen with 5 bit planes, no title. If the function fails then ADDRESS will be returned containing zero. If the screen is opened successfully, the ADDRESS contains the address of the start of the screen data structure. This is very much in the style of the library system. Similarly pointers to data are often used in functions rather than the inclusion of the data.

At the risk of making programs less legible, some functions have been compacted into simple but non-standard commands stolen from assembler. Instead of using X = X + 1, for example, you can use INC X. Or you can use bit shifting to multiply or divide. For example, 87 LSL 3 is faster than 87*3. There are many others. On the whole I rather approve of these functions since they offer the ability to maximise the speed of your code and they are a move towards assembler. Purists may, of course, differ.

So how does the language compare? Since all of you have copies of AmigaBASIC, I ran a few benchmarks to compare the two.This approach can be confusing and care must be taken to ensure that results are not masked by the effects of slow floating point operations or other non-constant factors. I haven't fully compared the performance of the two Basics, but it was certainly evident that F-BASIC was a real scorcher. First I tried a simple loop which performed six transcendential or floating point operations 2500 times. AmigaBASIC took 28.2 seconds compared to 3.24 seconds for F-BASIC. Integer operations were even faster. F-BASIC took 4 seconds to calculate the first 1259 prime numbers compared to a pedestrian 8 minutes by AmigaBASIC. A demo program using an IFF screen gave a performance comparable to similar programs written in C. The IFF file was decoded and loaded in an impressively fast time.

Regrettably there are omissions. Random access is not supported nor is direct loading of IFF files or multitasking. I believe that future versions will rectify some of these omissions. Overall I consider F-BASIC to be a significant improvement over AmigaBASIC and a cheap alternative to C. The language is well filled with useful commands and is generally conducive to structured programming. My main reservations are that it's a little tedious in the way that the editor/compilation cycle works and the error trapping seems to be a little awry. Additionally the manual, while giving loads of detail relating to commands, doesn't give enough help on the use of system Libraries or direct interfacing to the Amiga. The 60 odd examples supplied with the system do, however, go some way towards easing this short-coming.More than once an error in calling a function results in the dreaded Guru Meditation. However if you are looking for a language with a little more clout and have the patience to try something new, at £70 it's a fair buy. *YA*

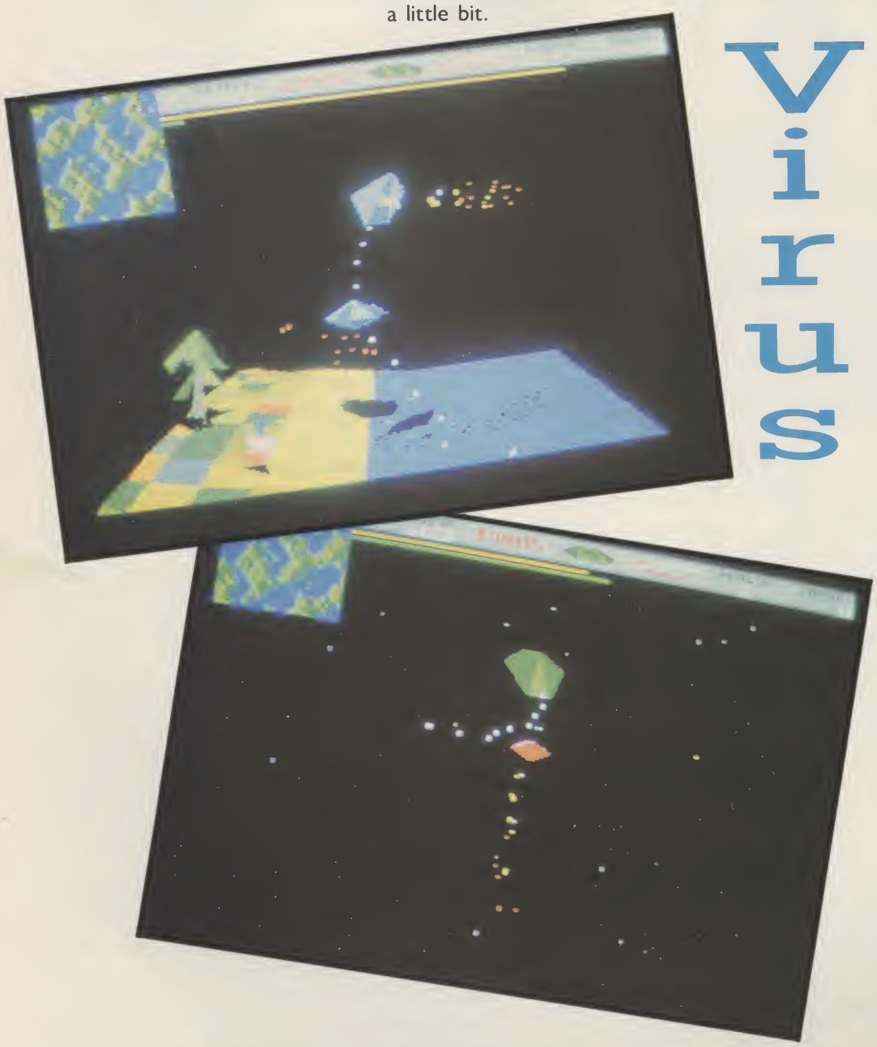
The famous Archimedes game comes down to the 16-bit world. Karen Young activates the smartbombs and lets fly

Virus

Quite possibly the most impressive demonstration of a computer system ever devised was the game called *Zarch*, a fairly basic shoot-em-up that, like so many games for Acorn machines took a lifetime to play, and offered very little to shoot at.

Being the only game for the Archimedes at that time, *Zarch* was, of course, a runaway hit, and it set a new standard for computer games: solid 3D representations as opposed to the trite wireframe images of *Star Wars* and *Starglider*.

Zarch has been moved down from the sedate 32-bit world of the Archimedes to the exciting and fast moving world of 16 bits. David Braben, author of *Elite*, the best selling game for the BBC Micro, converted the program and rewrote *Zarch*, but this time, adding a few bells and whistles to make sure the Amiga's sound sampling facilities are at least used a little bit.



So is the game a faithful reproduction of the original? Is the gameplay any better? Has *Virus* worked out whether it is a game or just an expensive demo? I am afraid that the answer is "no" to all three.

The problem is that *Virus* is not a faithful reproduction of *Zarch*. There are none of the wonderful shades of colour of the Archimedes' version, the scrolling (pixel by pixel on the Archie) is done by multiple pixel shifting (so the picture is jerkier than the Archie's) and there are more shadows on the surface of the planet in *Zarch*.

The gameplay isn't improved by using the mouse, although if you should want to, you can use the keyboard and a whole new aspect of the game comes to light — controllability.

I would have given this game one point (out of politeness) under the "playability" section of the pie chart at the end of this review, save for the fact that the keyboard control (not available on the Archimedes) is so much better than controlling your ship via mouse movements. There is a very useful key that automatically "de-inverts" the ship should you get into a spot of trouble, and this one key has helped save a lot of time reaching old high scores.

There are no "smart bombs" as there are on the Archie version, but the game does have the legendary sea monster, new scenes, flying fish (these can go a long way inland!), and a delightful sploshing sound of water as you fly over lakes and rivers. I reckon it must have been recorded during bathtime at Braben mansions as there is a distinct echo to the slopping and slapping sound of water. A nice effect though.

If you haven't seen the game running on *any* machine, let alone the Amiga, then you are in for a treat. The screen movement — while jerkier than the Archimedes' — is smooth by any other standards. The sound effects, while minimal, are still just right for the game, and they all add to the ghostly feel of the dying planet.

Why is this planet dying? Because of an alien race of Seeders, Drones and Mutants hell bent on infecting it, turning the soil red and replicating at annoying speeds. The Seeders fly over the planet spreading their spores which only lightly infect the ground. The real damage is done when they land on an area of dry land — this way they can spread seeds far more effectively.

You have three missiles that home in on the nearest thing in sight, and a laser gun. You do actually lose points for shooting and not hitting anything which turns the game away from a shoot-em-up, into a stake-em-out-and-shoot-em-up. If you hit trees, then you lose points, and if you hit the sea monster then you lose lots of points! Ecologically sound stuff and they can have my Greenpeace card any day.

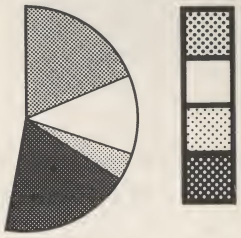
This is a difficult game to review, not on technical merit, because it really shows how good 16-bit games can be (and, by contrast, how much better the Archimedes version really is!) but on gameplay. I repeat the question — is this a game or just a very flashy demo? I am still not sure. The weird spaceships are all there in traditional Elite-style, but I cannot say that *Virus* is something I can get excited about. Naturally it will get the boring old coots who own Archies going, but then *any* game for their machine will please them. Sadly, though, I must say that the Archie version is better.

No gameplay really other than the odd thing to shoot at — otherwise very much a Defender clone but with the added novelty of the third dimension. *YA*

VIRUS

Title: **Virus**
Supplier: **Firebird**
64-67 New Oxford St
London WC1A 1PS

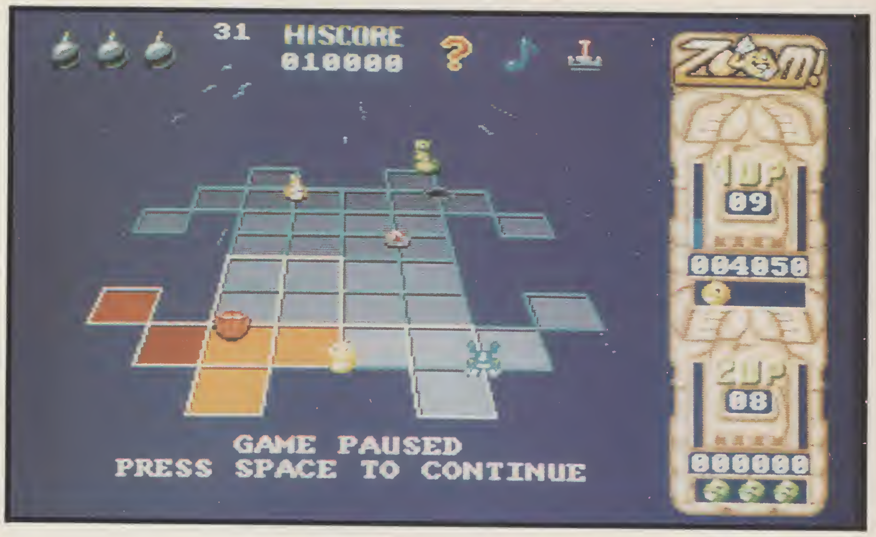
Tel: **01-379 6155**
Price: **£24.95**



Graphics: **20**
Sonics: **10**
Gameplay: **4**
Value: **19**

Zoom

Gordon Hamlett goes round in circles trying to fill in the squares



Isn't it always the case that the simplest games to play are inevitably the most addictive? *Zoom* from Discovery Software International is such a simple idea, you kick yourself for not having thought of it first.

Not that the idea itself is exactly original. You chase like mad round a grid trying to fill in all the squares before getting caught by the meanies (Very painful!) But the presentation of the game and its implementation combine to give a polished product.

There are fifty different screens to complete, each offering a different grid pattern and collection of pursuing monsters. Each grid consists of up to sixty four squares and all you have to do is to run completely round each one of them. As you move, so you leave a trail behind you. When you completely run round a square, it changes colour and can't be touched.

But running round each square individually is not the quickest way of completing the grid, as you have to retrace your steps too often. Together with the fact that you are being chased, you tend to leave lines all over the place, returning to complete the squares as and when you can. All this is food and drink to the Wormletts, one of a group of marauding monsters. Their particular delight revolves round erasing the lines that you have spent so much effort drawing. The grids are invariably designed full of holes so that you are constantly having to change direction or keep going round in circles in order to avoid one of the monsters

Other critters to be avoided include Jaggernauts, a set of deadly flapping lips, fast moving Spheroids and malicious Angleheads. As you progress through the levels, so black holes start to appear randomly. Walking into one is an excellent method of attaining instant oblivion status.

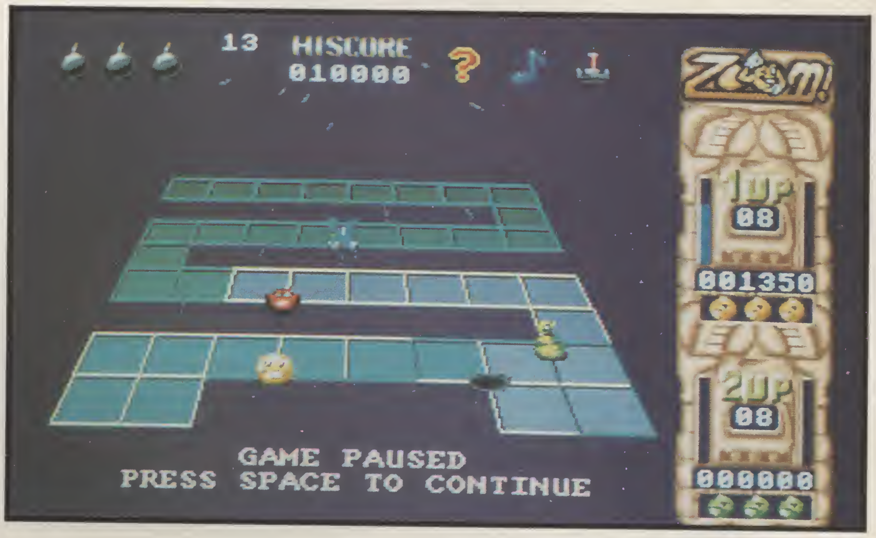
Not all is bad on the grid though and there is a variety of goodies to help your cause if you are fast enough to pick them up. Tubes of glue slow the enemy down whilst ice cubes temporarily freeze them completely. Candy speeds you up and magic potions render you invincible, at least until a timed gauge runs out. Apples complete four squares for you just like that and rockets zoom you off to the next level. Extra points can be scored via the money bags and as always, there is a joker in the pack. The question mark can do any of the above or kill you instead.

A final line of defence is to drop bombs behind you. These will slow down any creature that walks into it but be warned! Should you have to backtrack, you might find yourself dashing headlong into your own trap.

The game can be played either individually, taking turns with someone else or the two of you playing simultaneously, each trying to complete more squares than his or her opponent. In all versions, bonus points are awarded if you can complete more than one square at a time and this may or may not affect your strategy depending on what state of panic you happen to be in at the time.

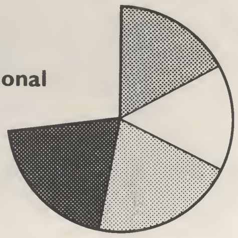
The game is well presented although it is not the sort of subject that lends itself to stunning graphics. You can choose to start at any of the first ten levels which seems a bit daft. A system of passwords allowing access to higher levels or increments of five when selecting the level would appear to make more sense. There is a jolly little introductory demonstration which looks like a song and dance act from the London Palladium. Unusual but of no relevance to the game and something that you are only likely to watch once.

Zoom is never going to set the games world on fire but it is addictive and the sort of game that is ideal for frittering away the odd half hour. *YA*

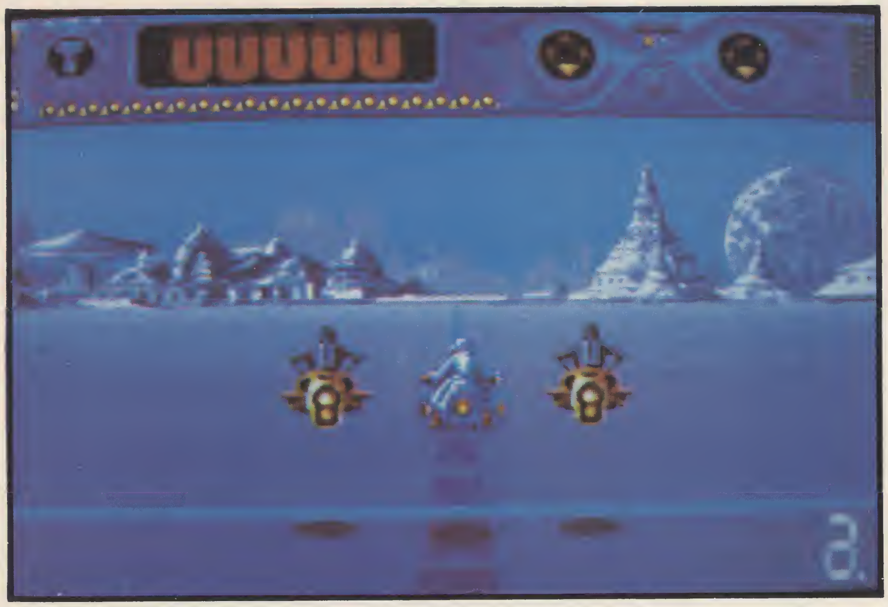


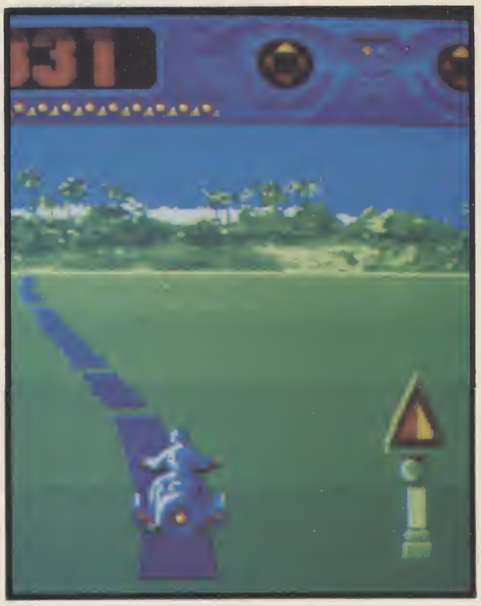
Zoom

Title: **Zoom**
Supplier: **Discovery Software International**
Price: **TBA**



▩ Graphics **17**
☐ Sound **15**
▨ Gameplay **21**
■ Value **20**





Space Racer Space Racer Space Racer

Kevin Crosby dons his crash helmet for a race game with some nice touches

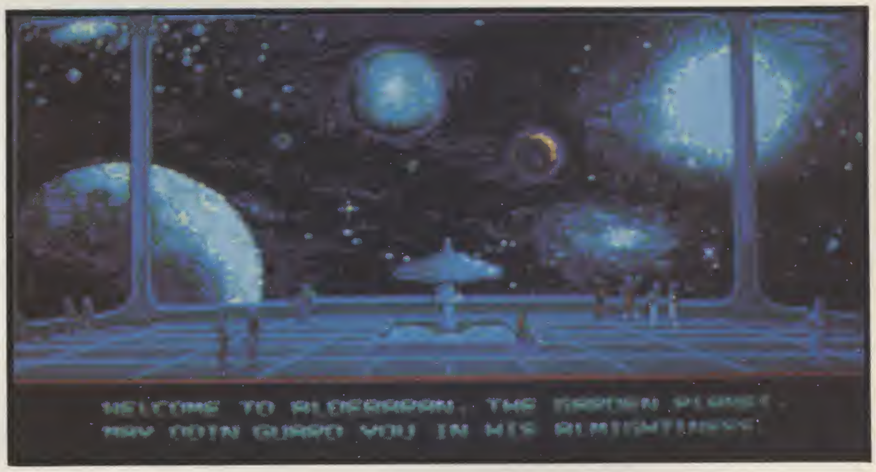
"Oh no!", I hear you say, "Not another race game". Fear not, this one does have a few saving graces. Allow me to set the scene...

We are now in the year 2132 where it would appear that the population of the universe is so bored that it has turned to violent Roman-style sports for enjoyment (obviously no copies of *Your Amiga* left to read!).

One such sport is the *Space Race* in which players must fly around a course on jet bikes avoiding obstacles and opponents. Of course it's not quite as simple as that, thanks to a smattering of land mines and so forth to keep you on your toes (or whatever the jet bike equivalent is). To assist you, you have a front-mounted gun which is useful for blowing away other racers in front of you. This saves a lot of hassle in overtaking them. You also have a control panel at the top of the screen which shows score, energy level and gives you early warning of the mines etc.

What's really nice about the game is the attention to detail which hits right from the loading screen which features some nice music.

Once loaded, the game options are displayed. You have a choice of three different locations for your races which consist of countryside, city and skull backgrounds. Once you've selected which you want, the little jet cyclist stands by his bike, waves to you, and says "Hello" before mounting up. Then, the race is on...



WELCOME TO [illegible], THE GARDEN PLANET.
MAY ODIN GUARD YOU IN HIS ALMIGHTINESS

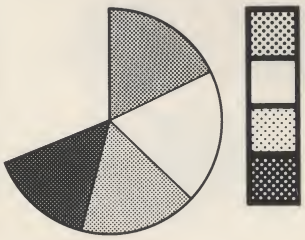
The action is reasonably fast but having seen some of the other title in Loriciels' range I can't help thinking that the company could have done a little better on the gameplay side. The gameplay is, however, made up for by nice humorous touches like the biker shaking his fist at obstacles he collides with.

Once you've had your fill of the three warm-up rounds you can go on to the championship phase. This is much the same as the previous rounds except the scores are logged on the high score table, which would be fine if they were saved onto the disk.

All in all a nice game but it didn't really hold my interest for very long. *YA*

SPACE RACER

Title: **Space Racer**
Supplier: **Loriciels**



Graphics: **18**
Sound: **19**
Gameplay: **17**
Value: **15**

YOUR AMIGA

01-437 0699
extension number 281

CLASSIFIED ADVERTISEMENTS

Lineage 58p per word (including VAT) —
Minimum £8.70
Semi-Display £10.80 (plus VAT) — Minimum 3×1

Send your requirements to:
Alan Cole, Classified Department,
Your Amiga, ASP LTD, Golden Square,
London W1R 3AB

VISA
Access

VIDEO GRAPHICS

WE SPECIALISE IN GRAPHICS, ANIMATION AND DTP SOLUTIONS

BUROCARE

Burocare Computer Systems
211 Kenton Road, Harrow,
Middlesex
Tel: 01-907 3636 Fax: 01-907 0027

TRIANGLE TELEVISION

TRIANGLE TELEVISION
The Amiga Graphics and Animation Specialists

FOR ALL YOUR VIDEO AND 'A.V.' REQUIREMENTS

Slide/Print/OHP Bureau Service. All systems are available and tested on our High Band Edit Suite. Ask about the Triangle Broadcast Genlock.

TRIANGLE TELEVISION
130 Brookwood Rd
London SW18 5DD
Tel: 01-874 3418 Fax: 01-877 1029

VISA
Access

MAIL ORDER

AMIGA P.D. LIBRARY

Public Domain Disks at £2.50 each

For free list send S.A.E. to
PO BOX 155
CARDIFF CF5 4YA

REPAIRS

AMIGA

REPAIRS AND SPARES — SEND YOUR FAULTY AMIGA FOR A FREE ESTIMATE TO:

OASIS

Dept. YA, 14 RIDGEWAY ROAD,
SALISBURY, WILTSHIRE, SP1 3BU
TELEPHONE (0722) 335061

REPAIRS

Fixed price repairs for Amiga 500's door to door service.

All approved modifications included free of charge.

Just £60 + VAT for door to door convenience.

GRANADA COMPUTER SERVICES,

Wilkinson House,
Telspace 2, Stafford Park,
Telford, Shropshire TF3 3BD

Tel: 0952 290488 — Ask for administration
VISA or cheque

THE COPY DEADLINE FOR THE FEBRUARY/MARCH ISSUE IS 1st DECEMBER

Tel: Alan Cole on 01-437 0626

FOR MORE DETAILS OF ADVERTISING

DISKS

ATHENE DISKS

85p EACH + VAT (98p)

Units of 10 Minimum Quantity 10
DS/DD 135tpc

Unbranded bulk discs by Maxell, Verbatim or KAO. Excellent quality. Full 2 for 1 guarantee.

Disc boxes, branded discs, Amiga A500, 2000
Please call for quantity discounts

VISA
Access

0705 511439 (24hr)
0705 511646 (Fax)

Cheques/PO to ATHENE DISKS,
16 Stoke Road, Gosport, Hants PO12 1JB

DO YOU HAVE A SERVICE TO OFFER YOUR AMIGA READERS?

IF SO ADVERTISE HERE

Why Take Risks?. . .
Buy **Branded** disks.

GOLDSTAR 3½" DS/DD DISKS
Boxed in 10s, 100% error free,
Branded, Fully Guaranteed,
Includes Labels

£8.30 + VAT Per box of 10

PLEASE ADD 87p + VAT FOR P&P
Goods sent same day as order received.

P.V. TUBES (MAIL ORDER DEPT).
104 Abbey St., Accrington, Lancs,
BB5 1EE
Tel: (0254) 36521/32611/390936
Fax: (0254) 395361
PARTNERS: Mrs S. Cucknell/
Mr B. Cucknell.

Access
VISA

SPECIAL OFFERS

T.C. COMPUTERS

TEL: Wantage 02357 60177

AMIGA A500	£369.99
AMIGA A500 & Colour Monitor	£610.00
AMIGA A500 & High Res Colour Monitor	£665.00
AMIGA System 500	£759.00
AMIGA System 500+	£2600.00
COMMODORE PC1 - mono	£369.00

Also Atari ST & PC Compatibles including Amstrad available

ENQUIRIES WELCOME
SAE for details to:
19 Havenvale, Wantage,
Oxfordshire OX12 7AJ

WEST MIDLANDS

WORDS Computers Christmas offer, A500, free television modulator, deluxe paint, only £365 including VAT and p&p. Amiga software also available. Tel 021-559 1857.

TERMS & CONDITIONS

CLASSIFIED ADVERTISING TERMS & CONDITIONS

Our terms for new advertisers (semi-display and lineage) are strictly pro-forma payments until satisfactory reference can be taken up (excluding recognised advertising agencies Cheques and P.O.'s should be crossed and made payable to ARGUS SPECIALIST PUBLICATIONS LTD and sent together with the advertisements to:

The Classified Dept.,
No. 1 Golden Square,
London W1R 3AB.

There are no reimbursements for cancellations. Advertisements arriving too late for a particular issue will be inserted in the following issue unless accompanied by instructions to the contrary. It is the responsibility of the advertiser to ensure that the first insertion of every series is published correctly, and corrections must be notified in time for the second insertion, otherwise the publishers will not accept liability or offer any reduction in charges.

All advertising sales are subject to Government Regulations concerning VAT. Advertisers are responsible for complying with the various legal requirements in force eg. The Trade Description Act, sex discrimination act & the business advertisements (disclosure) order 1977.

Full Terms & Conditions of Advertising available on request.

WE ACCEPT
ACCESS AND VISA
TELEPHONE BOOKING

DEALER GUIDE

TO ADVERTISE WRITE TO

Classified Dept
1 GOLDEN SQUARE
LONDON W1R 3AB

BERKSHIRE

CSE COMPUTERS
HORSESHOE PARK
PANGBOURNE
RG8 7JW
Tel: (07357) 4774
Contact Mr John Arkill
THE AMIGA SPECALISTS

CLEVELAND

MULTI COLOURED MICRO SHOP
6 Dundas Street, Redcar
Tel: (0642) 486643
OR
Unit 38/39, In-Shops,
Dundas Arcade, Middlesbrough
Tel: (0642) 230 323
We are Commodore approved Amiga specialists

ADVERTISERS

FOR DETAILS OF SERIES DISCOUNTS

Telephone:
01-437 0626

CORNWALL

AB & C Computers
THE SOUTH WEST AMIGA SPECIALIST
SHOWROOM & MAIL ORDER
Duchy House,
6 Lower Aylmer Square,
St Austell,
Cornwall PL25 5LQ
Tel: St Austell
(0726) 64463

HERTFORDSHIRE

ASK THE EXPERTS . . .
- ★ We stock a comprehensive range of Amiga Hardware, Software & Accessories
- ★ Wide selection of peripherals
- ★ Guaranteed media at competitive prices
- ★ Ruby-Comm Viewdate & VT100 Terminal software. The Complete Comms Pack for your Amiga or ST

Y2 COMPUTING
111 St Albans Rd, Watford WD1 1RD
Tel: 0923 50161

KENT

COMPUTER LEISURE CENTRE
117 High Street,
Orpington BR6 0LG
Tel: 0689 21101
Large range of 16-bit software in stock
Plus mail order service.

LONDON

AMIGA A2000 SPECIALIST DEALERS
Video and interactive video programmers and consultants
ARIADNE SOFTWARE LTD
273 Kensal Road,
W10 5DB
Tel: 01-960 0203
Contact: C West

GREATER MANCHESTER

COMPUTATILL LIMITED
77-79 CHADDERTON WAY
OLDHAM OL9 6DH
TEL: 061-652 8006
WE CATER FOR ALL YOUR AMIGA REQUIREMENTS

GREATER MANCHESTER

MIGHTY MICRO
Sherwood Centre,
268 Wilmslow Road,
Fallowfield,
M14 6WL
Contact Steven
Tel: (061) 224 8117
COMMODORES PREFERRED EDUCATIONAL DEALER
PERSONAL EXPORT SPECIALIST

ADVERTISERS

DISCUSS YOUR ADVERTISING CAMPAIGN WITH ALAN COLE

SUSSEX

MDR MAN COMPUTER MACHINE **INTERFACES LTD**
CONTACT US FOR YOUR
AMIGA 2000 and 500
SOFTWARE AND HARDWARE
EXPERT ADVICE GIVEN
ALSO BESPOKE SOFTWARE
Little Bridge House, Danehill,
Haywards Heath, Sussex RH17 7JD
Telephone: Danehill (0825) 790294

EAST SUSSEX

GAMER PERSONAL COMPUTERS
71 EAST STREET
BRIGHTON BN1 1HQ
TEL 0273 728681
Full range of Amiga Hardware and Software in Stock

TO ADVERTISE IN YOUR AMIGA DEALER GUIDE SEND YOUR REQUIREMENTS TO

ALAN COLE
YOUR AMIGA
1 GOLDEN SQUARE
LONDON W1R 3AB
Tel: 01-437 0626

WEST YORKSHIRE

TRILOGIC
STOCKISTS OF
- COMMODORE
- PHILIPS
- STAR
- CITIZEN
- SBC

Full range available at cash & carry prices from our new showroom at:
UNIT 1, 253 NEW WORKS RD,
BRADFORD BD12 0QP
Tel: 0274 691115.
See our ad elsewhere in this issue for Amiga accessories

DEALERS

Do you specialise in Computers?
This space could be yours
Call Alan Cole on
01-437 0699
today

3½

Lifetime Guaranteed, double sided, double density, 135 TPI, bulk wrapped discs with labels

Quantity	Price
10	£8.99
20	£17.89
25	£22.29
50	£43.99
100	£84.99

5¼

Lifetime Guaranteed, double sided, double density, 48TPI, bulk wrapped discs with labels

Quantity	Price
10	£3.49
20	£6.50
25	£7.99
50	£15.50
105	£31.00

NEW BULK DISC PACKAGE DEALS!

3.5" Discs

40 Discs + 40 Capacity box	£39.99
80 Discs + 80 Capacity box	£74.99

5.25" Discs

50 Discs + 50 Capacity box	£17.99
100 Discs + 100 Capacity box	£34.99

LOCKABLE DISC STORAGE BOXES

3.5"

40 capacity	£5.49
80 capacity	£7.49

5.25"

50 capacity	£5.49
100 capacity	£7.49

Plastic Storage Box – holds 10 (State 3.5" or 5.25") 75p

ST or AMIGA?

Ring us NOW for our ultra low prices on these two leading computers

UP TO 35% OFF ALL 16 BIT SOFTWARE

e.g. Menace (Amiga) ....£13.97 Starglider II (ST/Amiga) £17.39

ALL PRINTERS NOW WITH FREE BOX OF PAPER

Branded discs: Sony, TDK, Verbatim, Dysan 3.5" Discs
FULLY GUARANTEED MEDIA – BEST PRICES IN UK!

CHEAP! CHEAP! 3.5" DISKS 3.5"

10	£7.30
25	£18.00
50	£35.50
75	£52.50
100	£69.00

UNCERTIFIED, UNGUARANTEED, 3.5" BULK DISCS

Postage: All orders under £15.00 add 75p, Overseas £3. Over £15.00 FREE
All prices inc. VAT

HOW TO ORDER

CESTRIAN SOFTWARE

52 Brook Street, Chester CH1 3DN

Tel: Chester (0244) 312666 (3 lines)

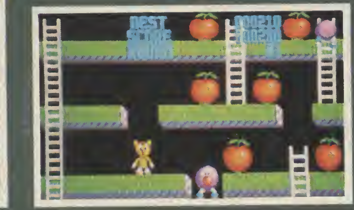
BY PHONE 0244 312666
BY POST TO ABOVE ADDRESS
BY FAX 0244 314635
BY TELEX 61556 CARIBA G

UNBEATABLE PRICES

ROBBEARY

Bertie, an agile and clever bear has targeted a famous 24 floors store for his last and most daring ROBBERY. With no alarms or visible guards and fabulous treasures, yet it has been avoided like the plague by the criminal fraternity. Bertie soon discovers why?

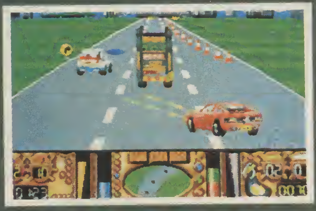
AMIGA £19.95



HIGHWAY HAWKS

Grand Prix driving skills is essential to negotiate the crowded highway at speed. Obliterate the assassins cars and the ones that get in your way but keep the tiger in your tank fed, the engine cool and the tyres and steering intact. Acquisition of faster cars and lethal weaponary depends on your driving and trading skills.

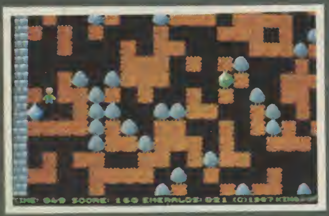
AMIGA £19.95 (2 DISCS)



EMERALD MINE

A giant arcade adventure which has received rave reviews. Each level has its own unique solution and requires ingenuity and dexterity to complete. One player or two players TEAM action for added enjoyment.

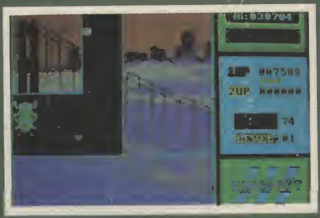
AMIGA £14.95
CBM64-PLUS 4 £7.95 (D) £9.95



MANIAX

Maniax relentlessly pursued by the creature undertakes to clear the thick fog engulfing the world capitals. An addictive game requiring quick thinking and action.

AMIGA-ST-IBM £19.95
CBM64-PLUS 4 £7.95 (D) £9.95



STRIP POKER II PLUS

A sizzling evening with
Sam & Donna

AMIGA-ST-IBM-ARCH. £14.95
SP-AMS-MSX-BBC-ELECTRON
CBM64-PLUS 4 £7.95

QUANTOX

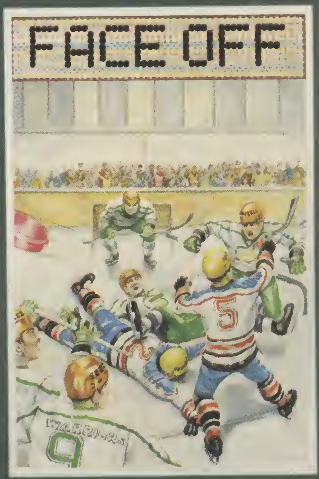
Progress through 24 levels of this fast and furious action. Lightning responses are secondary to the strategic choices between better weaponry or better defence.

AMIGA £14.95

FACE OFF

Experience the sheer pace and exhilaration of ICE HOCKEY. Be prepared for the body checks, fouls and rough play. League competition, 1 or 2 player option.

AMIGA-ST-IBM £14.95



STRIP POKER II PLUS DATA DISCS

Requires Strip Poker II Plus disc to load

Disc 1 Beverly & Dawn
Disc 2 Lee & Roy
Disc 3 Suzanne & Bianca
Disc 4 Rachel & Kim

AMIGA-ST £9.95

DAWN

MICRO TEXT

MICRO BASE

Ideal for home and business use. Menu driven to enable a novice to use powerful capabilities with minimal reference to the tutorial manual.

Micro base - a powerful data base with fast SEARCH and SORT facilities includes a very flexible label printer. Sorted files can be used by the Micro Text word Processor to send personalised letters. A boon to any Club Secretary.

AMIGA £19.95 EACH

BEVERLEY

ANCO

ANCO SOFTWARE LTD, UNIT 9-10 BURNHAM TRADING ESTATE
OFF LAWSON ROAD, DARTFORD, KENT DA1 5BH TEL: 0322 92513
MAIL ORDER HOTLINE: 0322 522631 FAX NO: 0322 93422
PAYMENT BY CHEQUE, P.O., VISA OR ACCESS